VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Nástroj pro analýzu a filtrování nežádoucího provozu

A Tool for Analyzing and Filtering Unwanted Traffic

2019 Bc. Jan Novotný

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jan Novotný**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 1801T064 Informační a komunikační bezpečnost

Téma: Nástroj pro analýzu a filtrování nežádoucího provozu
A Tool for Analyzing and Filtering Unwanted Traffic

Jazyk vypracování: čeština

Zásady pro vypracování:

Práce je zaměřena na zvýšení bezpečnosti webových serverů, a to pomocí implementace řešení pro identifikaci a filtrování nežádoucího provozu. Student se zaměří na identifikaci takových pravidel, které budou schopny filtrovat zejména provoz útoku typu zamezení služby (Denial of Service). Práce se bude skládat z teoretické a praktické části. V rámci teoretické části provede student průzkum aktuálních řešení v oblasti ochrany před útoky typu zamezení služby. V praktické části pak student provede implementaci řešení, které bude schopno analyzovat síťový provoz zachycený na serveru. Tento provoz pak bude filtrován na základě automaticky generovaných pravidel. Student provede experiment, který prokáže funkčnost řešení a míru úspěšnosti filtrace provozu.

Hlavní body zadání:
1. Rešerše aktuálního stavu na poli filtrování nežádoucího provozu.
2. Implementace řešení, které bude schopno analyzovat a filtrovat síťový provoz zachycený na serveru.
3. Testování řešení ve studentem připraveném prostředí.
4. Závěrečné zhodnocení experimentu a prezentace výsledků.

Seznam doporučené odborné literatury:

[1] P. Prasad, Mastering Modern Web Penetration Testing, Packt Publishing, 2016, ISBN: 978-1785284588
[2] J. A. Ansari, Web Penetration Testing with Kali Linux - Second Edition, Packt Publishing 2015, ISBN: 978-1783988525

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Plucar, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 26. června 2019 .....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 26. června 2019 .....

Chtěl bych poděkovat všem, kteří mi s touto prací pomohli a hlavně firmě NetDirect s.r.o, která mi umožnila použít firemní vybavení pro účely této diplomové práce.

Abstrakt

Útoky typu odepření služby(DoS/DDoS) jsou dnes stále velkým problémem webových aplikací a serverů. Tato diplomová práce je zaměřena na analýzu a filtraci těchto útoků. V teoretické části se zaměřuji na možnost detekce a filtrace těchto útoků a možné způsoby obrany. V praktické části prezentuji způsob řešení pomocí reverse proxy serveru, IPtables, ELK stacku a File2ban. Tento reverse proxy server monitoruje a filtruje síťový provoz směřem k webovému serveru. Tento provoz vizualizuji pomocí ELK stacku, který obsahuje ElasticSearch, Logstash a Kibana. Na závěr uvádím zjištěné výsledky a budoucí možnosti rozšíření obrany.

Klíčová slova: DOS, DDOS, útok, obrana, Nginx, ELK stack, Reverse proxy

Abstract

Denial of Service (DoS / DDoS) attacks are still a major problem for web applications and servers today. This thesis is focused on analysis and filtration of these attacks. In the theoretical part I focus on the possibility of detection and filtration of these attacks and possible ways of defense. In the practical part I present the way of solution using reverse proxy server, IPtables, ELK stack and File2ban. This reverse proxy server monitors and filters network traffic towards the web server. I visualize this traffic using an ELK stack that includes ElasticSearch, Logstash, and Kibana. In conclusion, I present the findings and future possibilities of expanding defense.

Key Words: DOS, DDOS, attack, deffense, Nginx, ELK stack, Reverse proxy

Obsah

Seznam použitých zkratek a symbolů

DOS	–	Denial of service
DDOS	–	Distribued DOS
UDP	–	User Datagram Protocol
TCP	–	Transmission Control Protocol
HTTP	–	Hypertext Transfer Protocol

Seznam obrázků

Seznam tabulek

Seznam výpisů zdrojového kódu

1 Úvod

V dnešní době jsou webové aplikace, jako např. e-shopy a diskuzní fóra, velice rozšířeny a těchto aplikací vzniká čím dál více. Díky této skutečnosti se také objevuje daleko více hackerů, kteří se danou aplikaci snaží využít ve svůj prospěch, ať už narušením činnosti, získáním citlivých dat nebo zamezením přístupu legitimním uživatelům.

V této práci se primárně zaměřuji na útoky typu odepření služby(anglicky Denial of Service, DOS). Tyto útoky mají společný cíl a tedy znemožnit serveru plnit účel, ke kterému byl vytvořen a tím ho znepřístupnit pro běžné uživatele. Výsledek tohoto snažení často vede k nespokojenosti běžných uživatelů i celých společností, které dané aplikace na serveru provozují. Pro vzorový příklad můžeme použít např. e-shop. V případě, že je e-shop nedostupný zákazník nakoupí požadované zboží u konkurence, což má za následek velkou ztrátu pro danou společnost, která e-shop provozuje. Z důvodu reputace i finanční ztráty. Tyto útoky nejsou jen nástrojem pro zabavení hackerů. Motivace k těmto útokům je, od testování znalostí, až po politicky motivované útoky. Tyto útoky často útočníci využívají k svému obohacení. K útoku DoS/DDoS se nejčastěji využívají infikované uživatelské počítače pomocí malware, které bez vědomí odesílají nežádoucí požadavky na cílový server. Dohromady tvoří obrovskou sít, kterou může útočník použít ke svým účelům. Vzhledem k dostupnosti vytvořených programů, pro tento typ útoků a jejich velmi lehké ovladatelnosti, může vytvořit útok prakticky kdokoliv.

Cílem práce je vytvořit software, který dokáže analyzovat, detekovat a filtrovat útoky typu odepření služby a také zajistit zmírnění dopadu těchto útoku na dostupné zdroje serveru. V teoretické části této práce vysvětluji základní principy DoS útoků. Dále popisuji pohledy firem zabývajících se obranou proti těmto útokům. Následně popisuji jednotlivé útoky, způsoby detekce útoků a jejich filtraci.

V praktické části bude simulován reálný provoz na reálnou webovou aplikaci společně s různými typy DoS útoků a následnou obranou. Mým cílem je vytvořit reverse proxy server, který bude schopen provoz procházející přes něj zachycovat, filtrovat, a který se k webovému serveru propustí.

2 DOS/DDOS útoky

První DDoS útok je veden v září roce 1996 proti firmě Panix sídlem v New York City, která je nejstarším a největším internetovým providerem[1][2]. Útočník zahltil počítače Panixu pomocí 150 SYN paketu za sekundu, takže Panix nemohl odpovídat legitimním uživatelům. Jelikož byl SYN útok spojen s podvrhnutými IP adresami, nedaly se detekovat a ani filtrovat. Ošetření tohoto problému spočívalo ve vytvoření speciální struktury, která držela napůl otevřené spojení, dokud neobdržela ACK paket. Spojení, které takto neodpovědělo bylo, po uplynutí 94 sekund, zahozeno.

Denial of service (DoS) je škodlivý typ útoku, ve kterém se útočník snaží napadnout internetovou službu nebo webové stránky a znepřístupnit je pro běžné uživatele. Obvykle dočasným znepřístupněním nebo vypnutím dané služby na serveru[3]. Je založený na tom, že daná služba není schopná zpracovávat velké množství požadavků. Jinými slovy se jedná útok, který brání cílové službě vykonávat to, k čemu byla vytvořena.

Princip DoS útoku spočívá v tom, že se použije neošetřená chyba v softwaru dané služby, nebo např. použitím dostupné URL adresy, která vyvolává určitou práci na serveru. Tato URL adresa může využívat více hardwarových prostředků serveru. Vygenerováním a zasláním nadměrného množství požadavků na danou službu dochází, v lepším případě, k využití většího množství prostředků serveru na vykonání daných požadavků a tím zpomalení ostatních uživatelů. V horším případě k vyčerpání všech prostředků a pádu celého systému, kdy následně musí proběhnout restart celé služby nebo serveru.

Takovéto útoky jsou prováděny z několika různých důvodu, od zkoušení vlastních znalostí studentů, až po poškozování konkurenčních společností a tím jejich oslabení na trhu. Často je také využíván k demonstracím na politických serverech. V neposlední řadě může být také využit jako teroristická hrozba.

Jeden ze známý útok na Českou republiku proběhl v říjnu 2017 při sčítání hlasů v parlamentních volbách[4][5]. Útočníkům se však příliš nedařilo a jejich útok byl během několika minut zablokován a vše bylo vráceno do normálního stavu.

V souvislosti s DoS existuje ještě jeden termín a to DDoS(Distribued Denial of service). Jedná se o techniku DoS, při které útočník používá větší množství zařízení, a tím zvětší sílu svého útoku k napadení cílového serveru. Takovému seskupení zneužitých zařízení se nazývá "Botnet".

2.1 Botnet

Botnet je síť napadených zařízení např. počítačů a mobilů, které jsou připojeny do internetu. Kontrolu nad takovým botnetem má hacker[6]. Dané zařízení je použito k rozesílání provozu na cílový server, díky tomuto jsou útoky velice komplexní a je těžké od sebe rozlišit legitimní a nežádoucí provoz.

K napadení zařízení dochází velmi lehce a to např. stažením souboru z nedůvěryhodných stránek. Tento typ malware poskytne plný přístup k ovládání zařízení. Napadené zařízení se následně pokouší tento malware rozšířit k vytvoření dalších "botů", které následně může útočník použít k hromadnému útoku. Počet napadených zařízení v botnetu je přibližně od několika tisíc, až po více než milión. Mezi největší botnety patří aktuálně Mirai botnet[7]. Nejvíce známých botnetů je v USA , přibližně 37%. Česká republika je umístěna v top 10 zemí s 2.24% [8].

K odstranění celého botnetu je nutné odstřihnout "hlavu", tedy zjistit, kde má daný botnet svoje hlavní zařízení, ze kterého je ovládán. V případě jeho nalezení je celý vypnut[9] a infikované zařízení již nemůže být tímto botnetem použito k útoku. Pro konkrétní zařízení je nejlepším způsobem, jak se dostat z botnetu, reinstalace OS z bezpečné zálohy. Pro zjištění, zda je daný počítač součástí botnetu je možné využít stránky společnosti Kaspersky dostupné z *https://checkip.kaspersky.com/*. Osobně kontroluji běžící procesy na svém PC v případě, že sleduji slabší výkon. Podařilo se mi takto zjistil, že jsem součásti "mining"botnetu, pro těžení kryptoměn.

2.2 Motivace pro DoS/DDoS

DoS/DDoS útoky během poslední 20 let velice vyvinuly. Moderní útočníci využívají výhod cloudu pro přístup k velkému výpočetnímu výkonu k provedení enormního DDoS útoku. Důsledky DDoS útoku na společnost může mít dalekosáhlé následky, výraznou ztrátu příjmů, porušení ochrany dat, a poškození pověsti. Mnohé společnosti se však domnívají, že daná věc se jim nikdy nemůže stát a nebrání si svoje webové servery a aplikace před těmito útoky. Pochopením motivace DDoS útoku je prvním krokem k určení rizika[10].

1. Finanční zisk

   Nejčastější motivací pro útok typu DDoS je finační zisk. Útočníci jsou schopni vybírat výkupné a vydírají nechráněné webové servery a aplikace. Dané společnosti pod útokem musí zaplatit útočníkovi bitcony pro zastavení útoku.

2. Protesty

   Z mnoha nefinančních důvodů může útočník zahájit DDoS útok. Mezi nejčastější patří protest proti organizacím, jejichž ideologie neodpovídá ideologii útočníka. Skupina jako Anonymous je známá tím, že často tyto protesty organizují. Známým protestem je protest proti ACTA, tedy smlouvě, která má chránit duševní vlastnictví na internetu.

3. Skrytí jiného útoku

   Další motivací útoku DoS/DDoS je zaměstnání IT odborníků společnosti tímto útokem, aby mohli použit další cílené útoky bez povšimnutí. Většinou pro krádež citlivých dat. Computer Weekly zveřejnilo v říjnu 2016 článek o tom, že většina DDoS útoků vyústila v krádež citlivých dat.

4. DDoS jako služba

   V dnešní době se rozmáhá pronájem DDoS útoku jako služby. Útočníci do svých botnetů získávají nové zařízení pro možnost poskytnutní této služby. Pro vyvolání DDoS útoku si útočník nechá zaplatit ve formě bitcoinu nebo formou PayPaylu a tím se obohacuje.

2.3 DDoS služba

DDoS se v dnešní době poskytuje také jako služba, kterou si může na internetu objednat prakticky kdokoliv. Tento pronájem se dá relativně levně koupit. Např. DDoS útok trvající tři hodiny o velikosti 125 Gbit/s stojí přibližně 180€[11]. Díky této skutečnosti se každý může stát terčem útoku, ale každý ho také může vyvolat a cíleně někomu škodit. Často se takto využívá k vydírání koncových uživatelů, kdy je požadováno zaplacení určité částky, aby byl DDoS útok vypnut[12]. Odborníci ze společnost Kaspersky v roce 2017 studovali DDoS služby na černém trhu a určili, do jaké míry tento nelegální obchod pokročil. Zpráva úvádní, že útok stojí pouhých 7$ na hodinu, zatímco cílová společnost může skončit zrátou několikanásobně větší[13].

2.4 OWASP TOP 10

Open Web Application Security Project (OWASP) jedná se o komunitu, která zdarma poskytuje články, metody, dokumentace, nástroje a technologie na poli bezpečnosti webových aplikací.

Nejznámější ze článků je OWASP TOP 10. Jedná se o informační dokument pro zabezpečení webových aplikací, který obsahuje 10 typu útoků seřazených jednotlivě podle bezpečnostních rizik, které představují pro webové aplikace. Členy tohoto projektu jsou různí bezpečnostní experti napříč celým světem, kteří sdílejí své znalosti. Tento dokument zvyšuje podvědomí společností a osob a nabádá je k tomu, aby kontrolovali svoje aplikace na bezpečnostní rizika. DoS útok byl do TOP 10 jako samostatný útok umístěn naposledy v roce 2004.

2.5 Stav dnešní obrany proti DDoS

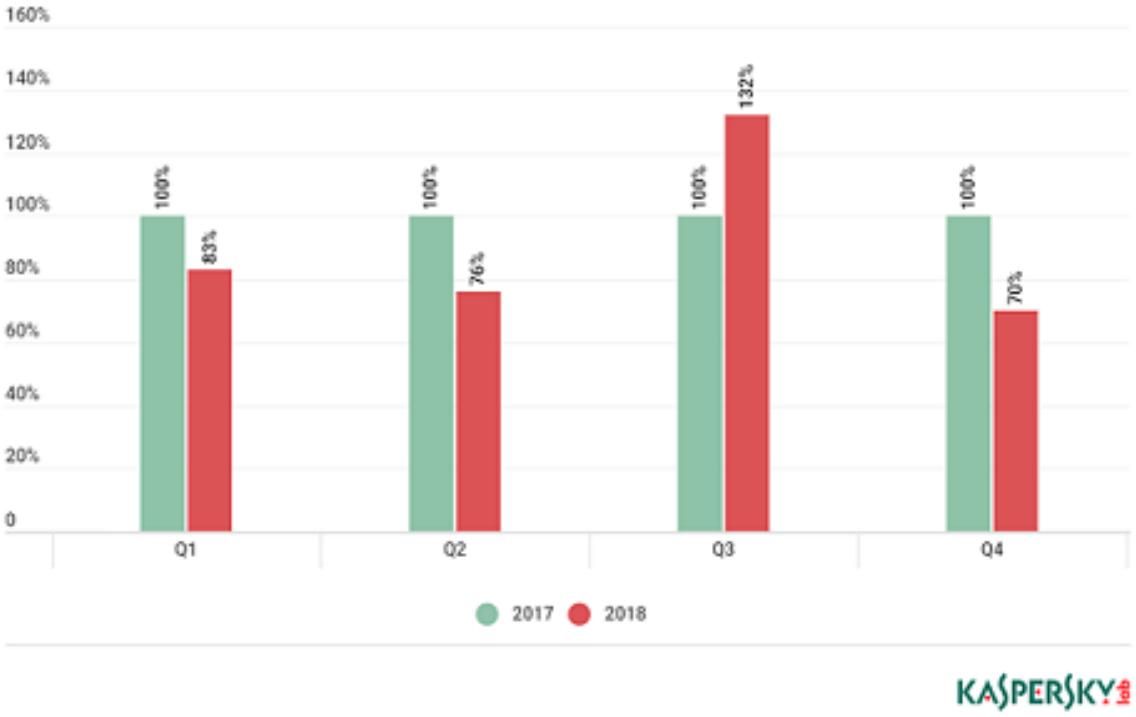
V dnešní době vzniká obliba cloudových služeb, kde se o obranu stará cloud service provider(CSP)[14]. Charakteristika DDoS útoků s různými přístupy a scénáři činí obranu proti DDoS složitou a je dobré využít jejich výpočetní výkon a způsoby obrany webových aplikací. Společnost Cloudflare poskytuje pro své uživatele obranu pomocí anycast sítě, kde je umístěno více než 116 datacenter[15] a celková šířka pásma činí 25Tbit/s.

2.6 DDoS útoky v roce 2018

V této kapitole přikládám jednotlivé reporty ohledně zaznamenaných DDoS útoků v roce 2018 pro vybrané společnosti Kaspersky, Akamai, NSFocus, Positive Technologies a Nexus Guard. Tyto společnosti pravidelně poskytují svoje poznatky na poli kybernetických zločinů.

2.6.1 DDoS report Kaspersky

Kaspersky Lab má dlouhou historii boje proti kybernetickým hrozbám včetně DDoS útoků všechny typů a složitostí. V roce 2018 bylo zaznamenáno společností Kaspersky o 13% méně DDoS aktivity než v roce 2017 [16]. Průměrná doba jednoho útoku se v roce 2018 zvýšila. V prvním kvartálu průměrně trval 95 minut[16]. Ve čtvrtém kvartálu průměrná doba útoku byla 218 minut[16]. Nejdelší útok trval rekordních 329 hodin. Jak je vidět níže v obr. č.1, ve všech kvartálech roku 2018, byla nižší četnost než v roce 2017 s výjimkou třetího kvartálu. Co se týče četnosti konkrétních DoS/DDoS útoků je s velkým předstihem nejpoužívanější SYN Flood útok, kde v žádném kvartálu roku 2018 neklesne pod 50% [16].



Obrázek 1: Porovnání kvartálů četnosti útoků Kaspersky 2017 a 2018

2.6.2 DDoS report Akamai

V roce 2018 byl zaznamenán přibližně stejný počet útoků DDoS jako v roce 2017 [17]. Mezi lednem 2017 a 2018 velikost DDS útoku vzrostla z původních 560Mbit/s na 783Mbit/s, nicméně celý rok 2018 zaznamenal zvýšení o 97.7% ve velikosti útoků s mediánem 0.56Gbit/s v lednu a 1.548Gbit/s v prosinci[17] .

DDoS by Quater							
2017 Q1	2017 Q2	2017 Q3	2017 Q4	2018 Q1	2018 Q2	2018 Q3	2018 Q4
1850	2354	2535	2348	2057	1845	2364	2142

Tabulka 1: Porovnání kvartálů četnosti útoku Akamai 2017 a 2018 zdroj[17]

2.6.3 DDoS report NSFocus

Report firmy NSFocus z roku 2018 uvádí, že bylo zaznamenáno celkem 14800 útoků což je o 28.4% méně než v roce 2017.[18]. Mezi první tři cíle útoků se umístili cloudové služby, hry a e-commerce. Nejčastějšími typy útoků byly SYN, UDP a HTTP, které společně tvoří 96% všech útoků [18], kde 13% útoků bylo použito v kombinaci s další metodou útoku.

2.6.4 DDoS report Positive Technologies

Společnost Positive Technologies v reportu z roku 2018 uvádí, že tento rok byl hlavně poznamenán novými rekordy ve velikosti útoků. Tyto útoky dosahovaly v prvním případě 1.35Tbit/s a v druhém případě 1.7Tbit/s[19]. První útok byl veden na hostující službu zdrojových kódů GitHub dne 28. února. Druhý útok byl veden na zákazníka poskytovatele služeb v USA. Další odhady uvádějí, že DDoS útoky budou silnější a častější z důvodu dostupného trhu s malware. Tímto budou i nekvalifikovaní útočníci schopni provádět útoky[19].

2.6.5 DDoS report NexusGuard

V reportech z roku 2018 společnost NexusGuard uvádí, zvýšení četnosti útoků typu UDP a TCP[20], dále uvádí zvýšení doby trvání útoků v průměru na 450 minut oproti minulému roku. Útok proti firmě Verge Network v druhém kvartálu způsobil ztrátu více než 1.7 miliónu dolarů[21].

3 Typy útoků

Typy jednotlivých útoků se rozdělují na tři základní kategorie. Volume Based Attacks, Protocol Attacks a Application Layer Attacks[22].

1. Volume Based Attacks

   Tato kategorie útoků se pokouší využít celou šířku pásma cílů, které zaplavují velkým množstvím dat. Tato kategorie zahrňuje např. ICMP Flood, UDP Flood a další, které používají falešné pakety. Tento typ útoků je velmi snadno realizovatelný díky velké dostupnosti nástrojů.

2. Protocol Attacks

   Cílem těchto útoků je využít všechny zdroje serveru pomocí chyby v síťovém protokolu. Do této kategorie patří např. TCP SYN Flood, Ping of Death, fragmentované útoky a další.

3. Application Layer Attacks

   Jedná se o útoky, které míří na aplikační vrstvu OSI modelu. Tyto útoky patří k nejtěžším na odhalení. Útočník se snaží vyčerpat všechny zdroje serveru webové aplikace pomocí velkého množství požadavků. Typickým zástupcem této kategorie je útok HTTP Flood.

Za zmínění stojí typy útoků s názvem Low and Slow Attacks[23]. Tyto útoky využívají velmi malou šířku pásma, na rozdíl od tradičních útoků. Tento provoz je velmi těžké odlišit od běžného provozu. Cílem těchto útoků jsou webové servery založených na vláknech s cílem využít všechny podprocesy s pomalými požadavky a tím znemožnit legitimním uživatelům přístupu k dané službě. Typickým zástupcem těchto útoků je Slowloris a R.U.Dead Yet?.

Na základě reportů společností jsem vybral 5 typů útoků, které měly velkou oblibu u útočníků při vytváření DDoS útoků. U každého útoku vysvětluji základní princip tohoto útoku a známe informace o něm.

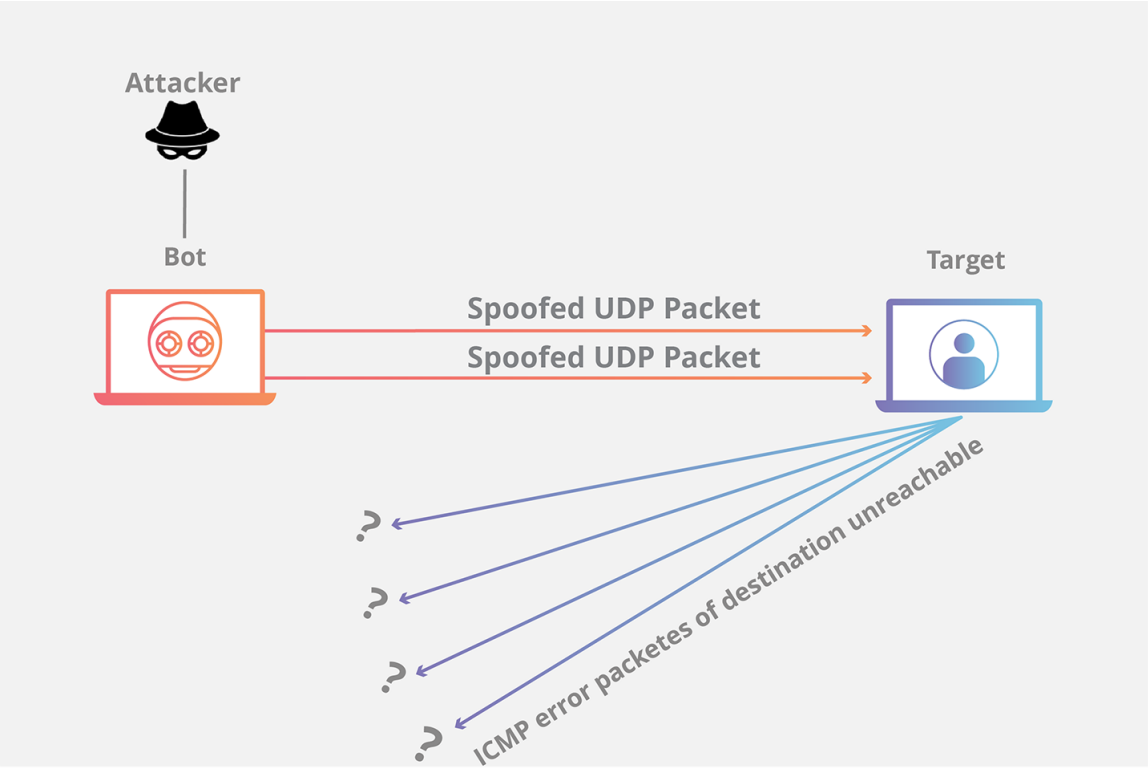
3.1 UDP Flood

UDP Flood je jedním z nejstarších zástupců DoS útoků. UDP protokol je bezstavový a tím pádem je útok velice prostý[24]. Cílem tohoto útoku je zaplavit náhodné porty na serveru. Využívá dvou základních věcí, které server musí udělat, aby daný požadavek vyřídil[25].

1. Musí zkontrolovat, zda na daném portu nějaká aplikace poslouchá.

2. V případě, že žádná aplikace na daném portu neposlouchá, server odpovídá pomocí ICMP "Destination unreachable"

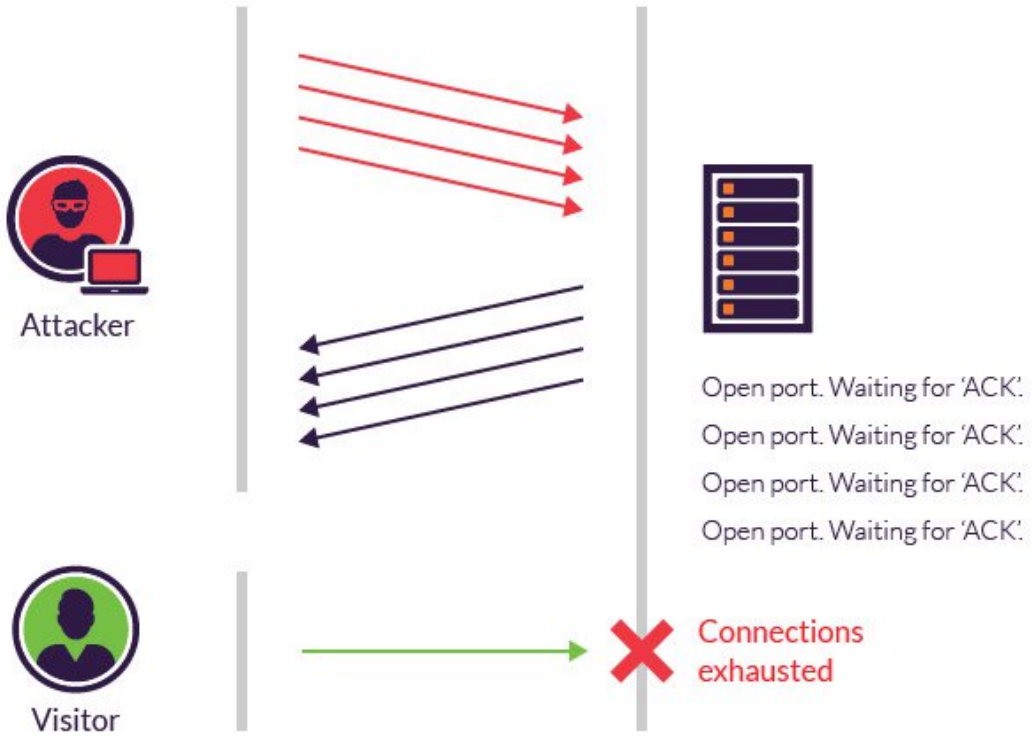
Společnost Cloudflare vysvětluje UDP útok tak, že ho přirovnává k hotelovým recepčním, kteří se starají o přepojování hovorů na konkrétní pokoje[25]. V případě zavolání musí recepční zkontrolovat, zda je daný host dostupný a popřípadě odpovědět, že není. V dnešní době se tento útok prezentuje průměrnou velikostí od 15.2Mbit/s až 290Gbit/s[26].



Obrázek 2: Princip útoku UDP Flood

3.2 TCP SYN Flood

TCP SYN Flood útok pracuje na transportní vrstvě OSI modelu[27]. Při běžném TCP "Three-way-handshake"provozu připojení začíná SYN paketem, který je odeslán klientem na server. Po tom, co cílový server paket přijme odpoví paketem SYN-ACK klientovi a přepne se do stavu SYN_RECV. Jako poslední klient odpoví na tento paket pomocí ACK serveru a spojení je vytvořeno[28][29]. Jedním z nedostatků TCP je udržování na půl otevřených připojení. Takto otevřené připojení je stav, ve kterém server čeká na potvrzení od klienta. Díky tomuto nedostatku útočník může vygenerovat větší počet těchto připojení a tím dojde k zahlcení serveru. Jelikož se překročí maximální počet vytvořených spojení, nedokáže server vytvářet legitimní připojení. Útočník generuje tyto pakety s podvrhlou IP adresou, z tohoto důvodu server nikdy neobdrží ACK paket a zůstává ve stavu *listening*[28]. Jako další varianta tohoto útoku je SYN-ACK Flood, kdy útočník na cílový server generuje velké množství podvrhnutých SYN-ACK paketů. V dnešní době se tento útok prezentuje průměrnou velikostí od 14.8Mbit/s až 359Gbit/s[26].



Obrázek 3: Princip útoku SYN Flood

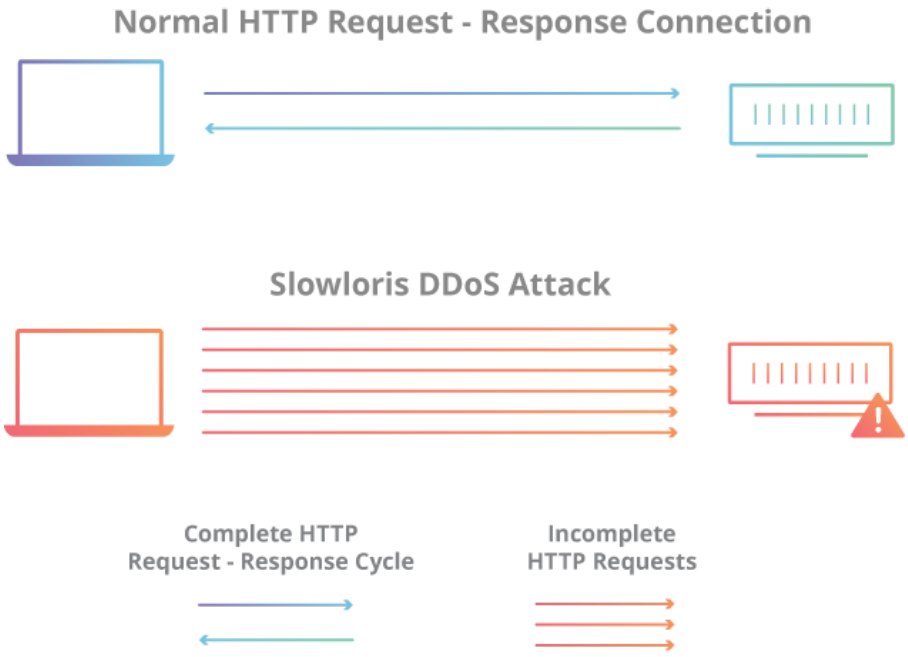
3.3 ICMP Flood

Internet Control Message Protokol(ICMP) slouží k získávání informací ohledně sítě. Vyšší protokoly jako TCP, dokáže zjistit, že paket nebyl doručen, ale ICMP zjišťuje závažnější problémy, jako je např. překročení Time To Live(TTL)[30]. ICMP Flood, také známý jako Ping Flood, využívá základní funkce ICMP ping. Tento příkaz slouží primárně k otestování připojení, zda lze komunikovat s jinými zařízeními v síti. Tomuto příkazu lze nastavit další parametry pro zvětšení paketu a rychlejšího generování[30]. Každý ICMP request musí server zpracovat a tím využívá svoje prostředky na to, aby mohl odeslat odpověď. Každá zpráva zabírá příchozí(echo-request) i odchozí(echo-reply) šířku pásma[31]. Vytvořením většího počtu takových požadavků dojde k přetížení pásma daného serveru. Tímto způsobem legitimní paket nikdy nedorazí k cílové stanici. V dnešní době se tento útok prezentuje průměrnou velikostí od 72.9Mbit/s až 137.9Gbit/s[26].

3.4 Slowloris

Tento útok je z kategorie pomalých útoků. Princip těchto útoků je ve využití malé šířky pásma. Díky této zkutečnosti je takový typ útoku mnohem složitější na odhalení[32]. Slowloris útok je zaměřen vždy pouze na jednu službu na konkrétním serveru. Všechny ostatní služby jsou tímto útokem nedotčeny.

Tento útok spočívá v tom, že se útočník snaží udržet co nejvíce současných spojení s daným serverem, po co nejdelší dobu[32]. Vytvoří připojení na daný server, ale pošle jen určitou část požadavku. Takto vytvoří velký počet požadavků, které nikdy nedokončí. Server nechává každé toto falešné připojení otevřené a čeká až bude kompletní. Vytvořením dostatečně velkého počtu připojení se překročí maximální počet současně vytvořených spojení, kdy server další nebude moci vytvořit pro legitimní uživatele. Hlavní výhodou tohoto útoku je, že nevyužívá šířku pásma dané sítě. Slowloris není efektivní proti všem webovým serverům např. IIS. Důvodem proč některé webové servery jsou chráněny již v základu, je způsob vytváření procesu ke konkrétnímu požadavku. V případě IIS spuštěné procesy čekají na kompletní požadavek, který následně vyřídí[33]. Podobnou variantou tohoto útoku je útok R U Dead Yet?(R.U.D.Y)[34]. Tento typ útoku vytvoří HTTP POST request, který je následně na server posílán velmi ve malých paketech o velikosti 1 bajtu a tím neustále drží otevřené připojení.

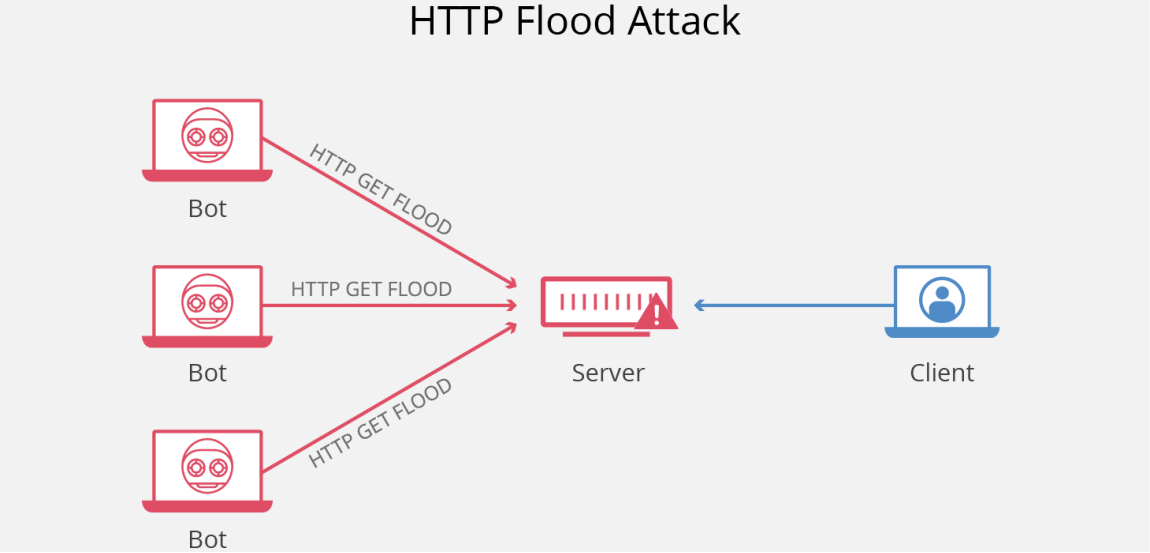


Obrázek 4: Princip útoku Slowloris

3.5 HTTP Flood

Tento útok pracuje na aplikační vrstvě OSI modelu, je tedy zaměřený přímo na konkrétní webovou aplikaci. Strategie tohoto útoku je využít co nejvíce zdrojů cílového serveru vytvářením velkého počtu požadavků typu HTTP-GET a HTTP-POST. Cílový server velice těžko dokáže rozdělit škodlivé od legitimních, protože škodlivý request je totožný, jako request od legitimního uživatele[36]. HTTP-POST request pravděpodobněji využívá více serverových prostředků pro zhotovení určitého requestu, jako např. výběr/zápis do databáze[37]. Vytvořením několika takových požadavků, musí server využít více prostředků, aby dané requesty odbavil. Při velkém

počtu již nebude hardware serveru zvládat odbavovat a bude nejčastěji request končit odpovědí 504 *time-out*. Principem tohoto útoku je tedy nasimulování velkého množství uživatelů přistupujících k webové stránce. Další variantou tohoto útoku je Single Request HTTP Flood [35]. Tento typ útoku obchází obranu, která je založena na základě velkého množství připojení. Útočník generuje maximální počet požadavků na webovou aplikaci tak, aby byl těmto ochranám neviditelný.



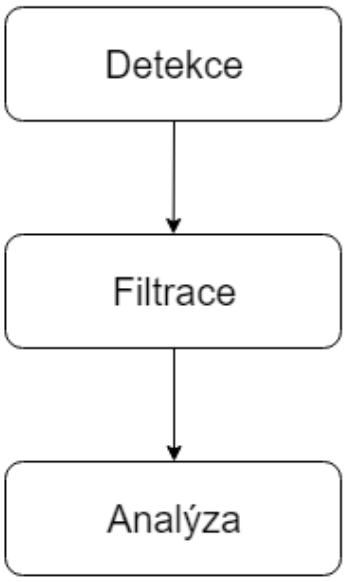
Obrázek 5: Princip útoku HTTP Flood

4 Obrana proti DoS/DDoS

Obrana proti DDoS útokům je velmi složitá s tím, jak se tyto útoky projevují[38]. Nejlepší způsob obrany je neustála analýza příchozího provozu a zamezením konkrétních přístupů dle vytvořených pravidel. Pokud je zjištěno, že je daná služba pod útokem, musí být obrana schopná rozdělit legitimní provoz od nežádoucího provozu. V případě, že detekce nežádoucí provozu vytvořená špatně, může zastaví také všechny příchozí provozy. Vzniká stejný problém, protože legitimnímu uživateli nebude umožněn přístup k dané webové stránce nebo službě. Zde nastává prostor pro vytvoření aplikace, která rozlišuje rozdíl legitimním a nežádoucím provozem a daný provoz blokuje. Níže popisuji konkrétní procesy, které musí aplikace splňovat, aby byla proti DDoS ochráněna.

4.1 Kategorie rozdělení obrany

Kategorie rozdělení obrany proti DDoS bych rozdělil do tří sekcí, kde první je detekce nežádoucího provozu, druhá filtrace nežádoucího provozu a třetí analýza provozu pro budoucí vylepšení obrany. Všechny tyto kroky se vzájemně ovlivňují a všem musí být kladen stejný důraz.



Obrázek 6: Fáze obrany proti DDoS

4.2 Detekce

Detekce je první krok při obraně proti útokům DDoS. V případě, že detekce není udělána správně, dle definovaných pravidel, je velice pravděpodobné, že nebude DDoS útok zachycen, nebo v horším případě bude zamezen přístup i velké většině legitimních uživatelů.

4.2.1 Detekce SYN Flood útoku

Způsob detekce TCP SYN Flood útoku může být vytvořena pomocí klasifikátoru. V případě, že paket ze stejného zdroje ke stejnému cíli přichází častěji, než je definovaná hodnota, můžeme paket označit za útočný a zablokovat ho[39]. Základní symptomem SYN Flood útoku je ten, že trvá delší dobu načtení webové stránky. V tomto případě, může být cílový server pod SYN Flood útokem. Detekce může proběhnout následovně.

```
netstat -tuna | grep :80 | grep SYN_RECV
```

Výpis 1: Příkaz pro zjištění SYN Flood útoku

Tento příkaz dokáže zjistit a vypsat IP adresy, které mají vytvoření spojení se serverem ve stavu SYN-RECV. V případě, že daný výpis ukazuje více připojení v tomto stavu, server může být pod SYN Flood útokem. Dané IP adresy z výpisu můžeme následně pomocí IPtables automaticky zablokovat.

4.2.2 Detekce HTTP Flood útoku

Základní symptomem tohoto útoku je razantní zpomalení nebo dokonce nedostupnost webové služby a nejčastěji server vrací odpověď 504 *time-out*. Detekce může být provedena kontrolou příchozího provozu. Např. když je příliš mnoho dotazů na konkrétní stránku webové aplikace, které mají stejný princip volání, je velice pravděpodobné, že aplikace je pod HTTP flood útokem. Detekce může být také vytvořena na základně běžného chování uživatelů na webu např. doba strávená na jedné stránce, počet požadavků během jedné session a mnoho dalších. Společnost Imperva pro ochranu aplikací používá výše zmíněnou klasifikaci všech příchozích požadavků. Na základě této klasifikace jsou příchozí požadavky povolovány/blokovány[40].

Další možností detekce nevyžádaných požadavků je využít funkčnosti prohlížečů a to pomocí javascriptu a cookies. V případě, že je požadavek dokončen pomocí prohlížeče, je možné vyvolat na server javascriptem informaci o tom, že byla stránka správně zobrazena.

4.2.3 Detekce UDP Flood útoku

Detekce UDP Flood útoku je velmi složitá. Pro zničtění tohoto útoku je nutné zaznamenávat příchozí UDP provoz, zda nepřechází do neúnosné hranice. Pokud je zaznamenáno více UDP paketů je možné, že server je pod UDP Flood útokem.

Detekci toho útoku je možno řešit pomocí databáze nevalidních IP adres, kterou si sami vytvoříme. Způsobů jak vytvořit takovou databázi je více. Jedním z nich je vytvoření počítadla IP adres, které se na dané služby dotazují. V případě překročení je taková IP adresa umístěna do databáze nevalidních IP adres a pro další dotazy zablokována.

4.2.4 Detekce ICMP Flood útoku

Detekce ICMP respektive ping útoku je velice jednoduchá. Defaultní velikost ping paketu je pro Windows 32 bajtů, pro Linux a MAC 56 bajtů[41]. V případě, že ping paket má větší velikost, pravděpodobně se někdo snaží použít nebo zjistit zranitelnost na tento útok.

4.2.5 Detekce Slowloris útoku

Základní symptomem tohoto útoku je malá vytíženost CPU serveru, avšak mnoho příchozích požadavků, které se aktuálně na serveru vykonávají. Tuto detekci je nutné provést přímo na serveru nikoli v jeho záznamech, protože IP adresa se do něj ukládá, jakmile je request dokončen. Detekce může proběhnout následovně.

```
$ netstat -ntu | awk '/^tcp/{ print $5 }' | sed -r 's/:[0-9]+$//' | sort | uniq
    -c | sort -n
```

Výpis 2: Příkaz pro zjištění Slowloris útoku

V případě, že se u dané IP adresy vyskytuje více požadavků, než je standardní, je pravděpodobné, že je aplikace vystavena Slowloris útoku.

4.3 Filtrace

Druhým krokem při obraně proti útokům typu DoS/DDoS, je filtrace těchto útoků na základě zjištěných informací z detekce. Základními možnostmi filtrace DoS/DDoS útoků je pomocí whitelistu, blacklistu a omezením příchozích požadavků. Níže popisuji možné způsoby filtrace jednotlivých útoků a úspěšnost daných způsobů.

4.3.1 Filtrace SYN Flood útoku

SYN Flood útok je již známý delší dobu a zajištění zmírnění dopadu je již vytvořeno.

Jako první možnost ošetření je zvýšení maximální počtu na půl otevřených TCP spojení. Každý OS má svůj vlastní definovaný počet takto otevřených spojení. Zvýšením tohoto počtu zmírníme dopad daného útoku, ovšem při větším útoku je tento způsob de facto zbytečný.

Jako další možnost je přepisování nejstaršího na půl otevřeného připojení jakmile je vyčerpán maximální počet takto vytvořených připojení. Nutnost tohoto opatření je ovšem to, aby legitimní připojení bylo vytvořeno rychleji, než bude dosažen maximální počet takto otevřených spojení. Tato obrana ovšem není také dostatečná při větším útoku.

Další možností filtrace SYN flood útoku je použití OpenFlow [29]. Tato funkce umožňuje flexibilní směrování paketů. Filtrace tímto způsobem je založena na použití základní funkce TCP a to RST bitu[42]. Při vytváření spojení je v první řadě použit SYN paket, na který následně server odpovídá SYN-ACK paketem se špatným potvrzovacím číslem[29]. Legitimní uživatel na

tento paket odpoví pomocí RST bitu. Takto ověřenému uživateli bude umožněna komunikace se serverem.

Jako další a účinná možnost je vytvořením SYN cookies[43]. Server posílá SYN-ACK rozšířenou o cookie a následně odstraní původní SYN paket od uživatele. Díky této cookie je následně server po tom, co mu přijde ACK, schopen opětovně vytvořit původní SYN paket a dokončit spojení. Tímto se dosáhne toho, že počet na půl otevřených připojení nedosáhne maximálního počtu.

4.3.2 Filtrace HTTP Flood útoku

Jak je již zmíněno výše, HTTP Flood útok je směřován na sedmou vrstvu OSI modelu. Útoky jsou často komplexní a mnohostranné. Jako jedna z možností obrany proti tomuto útoku, je použití uživatelského myšlení, na možnost odeslání konkrétního requestu, známého jako captcha. Typicky je to např. pro vložení příspěvku na fórech, diskuzích a e-shopech.

Další možnost ochrany proti tomuto útoku je vytvoření blacklistu IP adres, který je dynamicky naplňován konkrétní aplikací, nebo aplikací která funguje jako firewall.

Při tomto útoku záleží hlavně na schopnosti detekovat nežádoucí provoz, který je na server směřován. Případě, že není daný provoz zachycen, není co filtrovat.

4.3.3 Filtrace UDP flood útoku

Většina operačních systémů dokáže limitovat četnost odpovědí ICMP pakety a tím DDoS útok zmírnit. Nevýhodou tohoto ošetření je, že můžou být také filtrovány legitimní pakety.

Jako účinné opatření proti tomuto útoku je zakázat všechen UDP provoz mimo jasně definovaná místa např. konkrétní DNS záznamy v dané síti. Jedná se tedy o obranu formou whitelistu.

Společnost Cloudflare, pro obranu proti tomuto útoku, využívá výše zmíněnou obranu a tedy automaticky zahazuje všechen UDP provoz, který není spojený s DNS [45].

4.3.4 Filtrace ICMP flood útoku

Nejjednodušší obrana proti tomuto útoku je vypnutí ICMP funkce na daném serveru nebo routeru před ním. Ovšem tímto opatřením je nemožné používat všechny ostatní funkce ICMP jako je např. traceroute. Další možnost zmírnění útoku je již výše zmíněné omezení velikosti ping paketu na jeho defaultní hodnotu. Společnosti, poskytující cloudové řešení služeb, ICMP zachytávají pomocí anycast sítě.

4.3.5 Filtrace Slowloris útoku

Pro webové servery které jsou náchylné na tento typ útoku se dají použít možnosti rozdělené do tří kategorií.

1. Zvýšení dostupnosti serveru

Zvýšením počtu maximálního uživatelů zvýšíme útočníkův čas který musí vynaložit, aby daný server přetížil. Ovšem útočník vždy dokáže docílit maximálního počtu vytvořených spojení a tím je tato obrana ve většině případů nedostatečná.

2. Omezení příchozích požadavků

   Omezení na základě určitých faktorů pomůže zmírnit útok. Je možné omezit maximální počet požadavků vytvořených na jednu IP adresu, definování minimální přenosové rychlosti a omezení maximální doby po kterou má uživatel možnost zůstat připojený.

3. Využití cloudově založených služeb pro ochranu

   Cloudové služby, pro ochranu proti tomuto útoků, používají reverse proxy postavenou před webovou aplikaci, která se stará o průtoky směrem k serveru.

4.4 Analýza

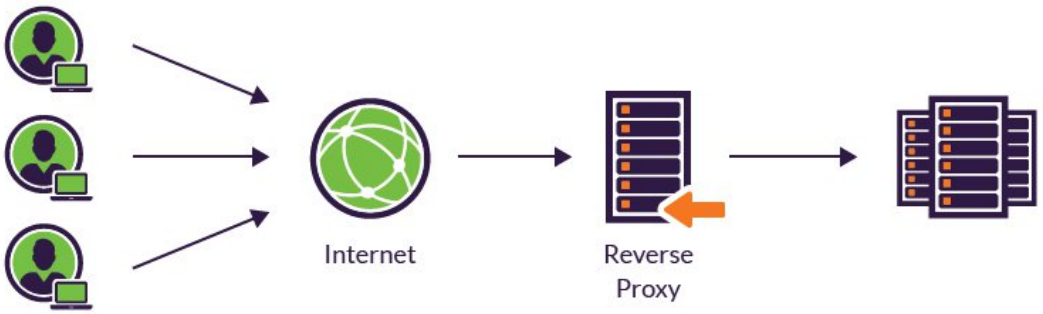
Jako prvním krokem při vytváření ochrany systému proti DoS/DDoS útokům je nutné, aby byly stanoveny hodnoty běžného provozu webové aplikace. Pokud bude nasazena ochrana proti těmto útokům bez provedené analýzy, je velice pravděpodobné, že zavedená pravidla zablokují většinu legitimního provozu společně s útokem, nebo nezablokují nic a tím pádem nebude obrana účinná.

Služba na serveru může být pod útokem, jelikož ji nedokáže detekce korektně zachytit. Z tohoto důvodu je nutné provoz na serveru podrobovat pravidelné analýze, ze které výstupem bude zjištění způsobu, jak daný útok prošel.

Je také nutné provádět analýzu detekčního systému, aby bylo možné stanovit kolik legitimního provozu bylo zablokováno a následně tento proces upravit tak, aby bylo docíleno menšího počtu zabránění legitimního provozu za minimální následky snížení funkce detekčního systému. Výstupem analýzy je tedy vždy způsob, jak vylepšit detekční systém, aby byl připraven detekovat nové hrozby a lépe již známé. Pro svoji analýzu provozu jsem použil ELK stack, který popisuji v praktické části.

5 Obrana pomocí reverse proxy

Reverse proxy je typ proxy serveru, který je umístěn před jinými servery a přesměrovává požadavky na příslušné servery dle příchozího požadavku. Když klient komunikuje s danou službou, je tato komunikace zpracovávána proxy serverem. Tímto způsobem žádný klient nekomunikuje na přímo s danou službou.



Obrázek 7: Reverse proxy

Výhody reverse proxy[44]:

1. Load balancing

   Webové servery, které jsou velmi využívané např. miliónem uživatelů denně, je nemožné, aby jeden server obsluhoval všechny tyto uživatele. Pomocí reverse proxy je možné distribuovat tyto uživatele na jiný webový server tak, aby nebyl přetížen. V případě, že je jeden server nedostupný je možné uživatele směřovat na dostupný do doby, než bude jiný server opět v provozu.

2. Global Server Load Balancing (GSLB)

   V této formě load balancingu, může být webová aplikace umístěna na více serverech okolo Země. Reverse proxy posílá požadavky na server, který je geograficky nejblíže. Tímto se sníží vzdálenost, kterou musí požadavky zdolat a tím se sníží doby načítání jednotlivých požadavků.

3. Obrana proti DDoS útokům

   Jelikož všechen provoz prochází přes reverse proxy, může tedy detekovat a filtrovat požadavky na základě zavedených pravidel a tím propouštět pouze legitimní uživatele, kteří mají cílený zájem danou službu využívat.

4. Cachování požadavků

Pomocí proxy serveru je možné cachovat obsah. Tímto způsobem je dosaženo vyššího výkonu odpovědí serveru. Např. pokud jeden si uživatel zobrazí hlavní stránku e-shopu, může být tato stránka proxy serverem cachována a další uživatel, který vyvolá stejný request je automaticky obsloužen proxy serverem bez využití webového serveru na kterém aplikace běží.

5. SSL šifrování

   Šifrování a dešifrování SSL(TLS) komunikace pro každého klienta je v součtu náročná operace pro webový server. Reverse proxy může být nastaven tak, aby dešifroval všechny příchozí dotazy a šifroval odchozí. Tímto je cílovému serveru ulehčeno a může použít zdroje pro svůj účel.

6. Skrytí intranetu

   Reverse proxy server je vždy postavený jako vstupní brána na hranici sítě. Všechen provoz je směřován přes něj. Pokud útočník chce napadnout server umístěný v síti, musí tedy projít přes reverse proxy server a tím nepozná vnitřní infrastrukturu sítě.

6 Obrana pomocí IDS/IPS

IDS a IPS jsou systémy pro detekci a monitoring sítě[46]. Hlídají předdefinované události, které by mohly být náznakem pokusu o útok a také jim dokážou předcházet. Tyto aktivity jsou následně zaznamenány. Na základně těchto záznamů je možné vyhodnotit nebezpečí.

IDS *(anglicky Intrusion Detection System)* je pasivní monitorovací systém síťového provozu. Na základě vzorů dokáže určit, které přístupy do sítě patři mezi neobvyklé. IDS nedokáže zabránit tomuto provozu, pouze dokáže informovat kompetentní osoby např. administrátory o neobvyklé aktivitě, kterou je následně nutné ručně vyhodnotit. IDS je ve většině případů postaven na velkém množství zdokumentovaných útoků. Na základě těchto vzorů hledá společné znaky již známých útoků. Podobně pracuje např. antivirový program, který software detekuje na základě databáze vzorů.

IPS *anglicky Intrusion Prevention System* je označován jako vyšší stupeň ochrany, jelikož dokáže tyto útoky zaznamenat a určit kroky pro jejich blokaci těchto útoků pomocí firewallu. Stejně jako IDS používá databázi známých útoků. V případě tohoto použití se jedná o přístup na základě vzorů (Signature-based). V případě, že útok se podobá definovanému vzoru je ihned zablokován. Další přístup je na základě anomálií (Anomaly-based) síťového provozu. V případě, že je taková anomálie detekována, systém ji automaticky zablokuje. Posledním přístupem je na základě ručně definovaných zásad (Policy-based). Tento přístup vyžaduje zásah administrátora pro nastavení daných zásad na základě bezpečnosti dané orgranizace.

Mezi často používané IDS a IPS systémy pro Linux patří Snort, Suricata a Fail2ban, který popisuji v dalších kapitolách.

7 IPtables

IPtables[47] je nástroj, který umožňuje linuxovému systému plně pracovat se síťovou komunikací. Pomocí něj je možné snadně nastavit firewall a tím řídit provoz na daném serveru. IPtables poskytuje, na základě zadaných pravidel, možnost povolovat, blokovat, limitovat a zaznamenávat příchozí i odchozí provoz. Při vytváření záznamů v IPtables je nutné věnovat pozornost pořadí pravidel, v případě, že paket splňuje definované pravidlo, je proces zastaven a není dál kontrolován ostatními pravidly. Existují 4 tabulky. Každá tato tabulka obsahuje základní řetězce, ale může také obsahovat uživatelsky definované řetězce.

1. Filter

   Tabulka Filter je výchozí tabulkou. V případě, že není definovaná jiná tabulka, je pravidlo automaticky umístěno do této tabulky. V této tabulce je možné použít tyto řetězce INPUT, FORWARD a OUTPUT.

2. Nat

   Tato tabulka je určená pro NAT. V případě, že je vytvořeno nové spojení, je tato tabulka kontrolována na zadané pravidla. V této tabulce je možné použít tyto řetězce PREROUTING, OUTPUT a POSTROUTING.

3. Mangle

   Mangle tabulka slouží ke kontrole hlaviček příchozích paketů. V této tabulce je možné použít tyto řetězce INPUT, FORWARD, POSTROUTING, PREROUTING a OUTPUT.

4. Raw

   Tato tabulka primárně slouží k označení paketů, které nebudou dále kontrolovány. V této tabulce je možné použít tyto řetězce PREROUTING, OUTPUT.

INPUT řetězec se použije pro příchozí připojení. FORWARD řetězec se používá v případě, když je žádoucí pakety, které splní dané pravidla, přeposlat. OUTPUT řetězec se používá pro odchozí spojení. PREROUTING řetězec se aplikuje při vstupu příchozího paketu na server. POSTROUTING řetezec se použije před odchodem paketu ze serveru. Nyní je nutné vysvětlit nejpoužívanější parametry, které se ve spojení s IPtables používají.

1. -p protocol. Tento parametr určuje, pro jaký protokol bude dané pravidlo použito např. tcp, udp, icmp. V případě, že není definovaný tento parametr je automaticky definovaný pro všechny.

2. -s source IP address. Pomocí tohoto parametru je možné určit, pro jakou IP adresu, popř. rozsah adres, bude dané pravidlo použito.

3. -d destination IP address. Tímto parametrem se určuje cílová IP adresa. Stejně jako u parametru -s je možné použít konkrétní adresu nebo rozsah adres.

4. -m match. Tímto parametrem se určuje, že budou dále testovány podmínky pro které bude dané pravidlo platit.

5. -j jump. Tento parametr určuje cíl, jak bude s daným paketem dále zacházeno. Díky tomuto parametru může být pravidlo směřováno na uživatelsky definovaného řetězce.

6. -i interface. Tímto parametrem se určuje, který interface bude daným pravidlem ovlivněn. V případě, že není zvolen žádný, pravidlo platí pro všechny.

IPtables poskytuje možnost kontrolovat příchozí a odchozí pakety pomocí parametru *-m* na základě definovaných podmínek. Je možné kontrolovat např. počet příchozích požadavků, stavy konkrétních připojení, velikost paketu a spoustu dalších. Níže sepisuji tyto základní podmínky pro manimulaci s pakety[48].

1. Length

   Tato podmínka kontroluje, na třetí popř. čtvrté vrstvě, velikost paketu oproti zadaným hodnotám. Tvar zadávání je následující: *–length xxx:yyy*. Touto podmínkou např. lze povolovat pakety ICMP pouze defaultních velikostí. Je možné tento parametr invertovat pomocí *!*.

2. Limit

   Tato podmínka sleduje počet paketů, které se k danému serveru přichází nebo z něj odchází. Díky této podmíce je tedy možné omezit počet těchto spojení. Používá se pomocí *–limit rate/[s/m/h]*. K této podmínce je dobré použít také podmínku *–limit-burst xx*. Tímto parametrem se nastaví maximální počet paketů např. 100. Tato hodnota je postupně naplňována každým příchozím/odchozím paketem. Podmínka *–limit* tento počet snižuje. V případě, že je naplňen maximální počet, je tato podmínka nesplněna.

3. Connlimit

   Tato podmínka funguje podobně jako *–limit* s rozdílem, že se měří počet připojení dle IP adresy. K této podmíce se připojují *-upto, -above, -mask, -saddr, -daddr*. *-upto* určuje maximální počet připojení do hodnoty *n*. *-above* bude splněna pokud počet připojení překročí hranici *n*. *-mask* určuje masku sítě, pro kterou bude podmínka použita např. 24. *-saddr* a *-daddr* určuje zdrojovou a cílovou IP adresu.

4. Conntrack

   Tato podmínka sleduje možné stavy příchozích paketů. Nejčastěji se používá ve spojení s *–ctstate*. Možné hodnoty jsou INVALID, NEW, ESTABLISHED, RELATED, UNTRACKED, SNAT a DNAT.

5. Multiport

Tato podmínka určuje konkrétní porty. Může být definováno až 15 portů. S touto podmínkou se používá *–source-ports, –sports* a *–destination-ports, –dports*. Je možné zadávat i rozsahy portů.

IPtables příkaz se standartně ukončuje použitím parametru *-j*, kdy se určí cíl, jak s daným paketem, který splnil dané pravidla, bude naloženo.

- ACCEPT Tento cíl přijme paket a již není kontrolován na další pravidla.
- DROP Použitím tohoto cíle bude daný paket zahozen.
- RETURN Tato možnost se používá v případě uživatelsky definovaného řetezce. Použítím tohoto cíle bude provádění vráceno na místo odkud se do daného řetezce paket dostal.
- REJECT Pomocí této možnost bude paket zahozen a současně je možnost navolit odpověď, která bude vrácena.
- LOG Tento cíl neukončí procházení paketu, pouze vyvolá zápis do logu systému.

Pro logování paketů je možné použít zmíněný LOG. S použitím *–log-prefix "value"* je možné přidat konkrétní prefix pro logovaný paket. Na základě této možnosti je možné v */etc/rsyslog.conf* nastavit pravidla pro logování záznamů z IPtables. V této práci používám logování pro přijaté i zahozené ICMP, TCP, UDP pakety.

8 Nástroje pro útok

Většina níže sepsaných nástrojů původně vznikla za účelem testování aplikace pro chování při velké zátěži. Po uvolnění začaly být využívány právě pro opačný cíl a to útoky. Pro svoji praktickou část jsem použil níže uvedené nástroje.

- HPing3

  Jednoduchá konzolová aplikace, která je inspirován z nástroje ping[49]. Může být využita pro generování velkého množství TCP, UDP a ICMP provozu. Dokáže také podvrhnout zdrojovou IP adresu. Pro zrychlení odesílání paketů se používají parametry *–fast –faster –flood*, kde *–flood* je nejrychlejší možný způsob.

- LOIC (Low Orbit Ion Cannon)

  Jedná se o velmi jednoduchý a nebezpečný nástroj, který byl proslaven skupinou Anonymous, která ho využila pro útok proti velkým společnostem jako např. FBI, PayPal, Visa, MasterCard[50]. Pokročilejší verze LOIC, která byla upravena skupinou Anonymous, umožňuje HIVEMIND mód[51]. Tímto této funkci se klient připojí k IRC serveru. Takto připojený klient může být ovládán vzdáleně a na příkaz útočit na webové stránky a vážně ji ohrozit.

  V grafickém rozhraní je potřeba zadat pouze IP adresu a zvolit způsob útoku TCP, UDP nebo HTTP. LOIC v základní verzi nedokáže podvrhnout zdrojovou IP adresu

- Slowloris.py

  Jedná se o program vytvořený v pythonu, který dokáže generovat Slowloris útok. Každých 15 sekund odesílá na cílový server hlavičky requestu, aby udržel spojení do doby, než ho ukončí server. Jakmile je spojení ukončeno, automaticky vytvoří nové. Umožňuje také využít útok pomocí proxy.

9 Reverse proxy pomocí Nginx

Na základně doporučení jsem pro vytvoření reverse proxy použil aplikaci Nginx, která je volně dostupná. Jedná se o webový server s možností load balancingu a reverse proxy. Dokáže pracovat s protokoly HTTP, SMTP, POP3, IMAP a SSL. Nginx byl vytvořen Igorem Sysoevem v roce 2004, pro vyřešení C10K problému. Tento termín vznikl v roce 1999 pro popis obtížnosti existujících webových serverů s velkým přístupem 10000 připojení. Díky asynchronnímu přístupu se Nginx stal nejrychlejším web serverem[52]. Tento web-server dnes používá přibližně 41% všech webových serverů[53]. Níže popisuji svůj postup při instalaci a konfiguraci použití Nginx, jako reverse proxy pro cílový server. Pro instalaci je nutné použít tento příkaz.

```
$ apt-get install nginx
```

Výpis 3: Příkaz pro nainstalování Nginx

Jakmile je Nginx nainstalován je nutné vypnutou defaultní nastavení pomocí níže uvedeného příkazu.

```
$ unlink /etc/nginx/sites-enabled/default
```

Výpis 4: Příkaz pro vypnutí defailtní nastavení Nginx

Dále je nutné vytvořit soubor libovolného názvu v adresáři */etc/nginx/sites-available/*. Pro svoje účely a přehlednost jsem použil *reverse-proxy.conf*.

Pro základní nastavení a otestování funkčnosti reverse proxy stačí níže uvedená konfigurace, která v základu udělá pouze to, že poslouchá na portu 80 a všechny requesty přeposílá na definovaný server. Tuto základní konfiguraci dále rozšiřuji o cachování, ručně definované logování, vlastní accesLog, errorLog a limitaci počtu požadavků. Výsledný konfigurační soubor je umístěn v příloze této práce.

```
server {
   listen 80;
   location / {
      proxy_pass http://www.diplomka.loc;
   }
}
```

Výpis 5: Základní konfigurační soubor reverse proxy

Nejdůležitější součástí v dané konfiguraci je direktiva *proxy_pass*, která určuje, kam je daný provoz přes reverse proxy směřován. Po vytvoření daného souboru je nutné daný konfigurační soubor aktivovat. Pro aktivování je možné použít tento příkaz. Nginx vždy konfigurace bere ze složky */etc/nginx/sites-enabled/*, je tedy nutné do dané složky soubor umístit pomocí níže uvedeného příkazu.

```
$ ln -s /etc/nginx/sites-available/reverse-proxy.conf /etc/nginx/sites-enabled/
    reverse-proxy.conf
```

Výpis 6: Příkaz pro vypnutí defailtní nastavení Nginx

Pro zjištění správnosti konfiguračního souboru je možné použít tento příkaz. V případě, že je konfigurační soubor v pořádku je možné službu Nginx restartovat.

```
$ service nginx configtest
$ service nginx restart
```

Výpis 7: Otestování a restart Nginx

10 ELK stack

Pro lepší použitelnost a analýzu provozu jsem se rozhodl zaznamenávat data pomocí ELK stacku[54]. Jedná se o velmi populární nástroj, který spojuje tři projekty do jednoho. Používá se pro vyhledávání, analýzu a vizualizaci v reálném čase. Tento nástroj používají firmy jako NetFlix, Stack Overflow, LinkedIn a další[55]. Skládá se z těchto komponent.

1. ElasticSearch

   ElasticSearch je open-source, RESTful distribuovaný vyhledávací nástroj založený na Apache Lucene. V dnešní době je tento program velice oblíbeným nástrojem pro fulltextové vyhledávání a analýzu logů. Je vyvinutý v Javě a lze sním komunikovat pomocí webového rozhraní. Do ElasticSearche se posílají data ve formátu JSON pomocí API nebo níže zmíněného Logstashe. Cílí hlavně na rychlost zpracování dat[56].

2. Logstash

   Logstash je open-source nástroj pro zpracování dat. Umožňuje sbírat data z více zdrojů najednou. Tyto data za běhu posílá do požadovaného cíle. Nejčastěji se používá právě ve spojení s ElasticSearch.

3. Kibana

   Kibana je open-source nástroj pro vizualizaci a průzkum dat. Slouží k monitorování aplikací. Nabízí výkonné a snadno použitelné funkce, jako jsou různé typy grafů včetně geolokace dle IP adres. Kibana se nejčastěji používá právě pro vizualizaci dat z ElasticSearche.

10.1 Instalace ElasticSearch

ElasticSearch je databáze, kde jsou uloženy data ve formátu JSON. Data pro ElasticSearch poskytuje Logstash, který je zpracovává. Instalace se provede pomocí těchto příkazů:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key
    add -
sudo apt-get install apt-transport-https
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee
     -a /etc/apt/sources.list.d/elastic-7.x.list
sudo apt-get update
sudo apt-get install elasticsearch
```

Výpis 8: Příkazy pro instalaci ElasticSearche

V danou chvíli je nainstalován ElasticSearch a je nutné ho nakonfigurovat. V konfiguračním souboru */etc/elasticsearch/elasticsearch.yml* odkomentujeme tyto řádky *http.port: 9200, network.host: localhost*. Touto konfigurací se nastaví konkrétní webový server, na kterém Elas-

ticSearch poslouchá zde *http://localhost:9200*. Pro vyšší výkon ElasticSearche je nutné přenastavit heap size[57]. Tuto konfiguraci je možné provést v konfiguračním souboru */etc/elasticsearch/jvm.options*. V defaultním nastavení je heap size omezen na 1GB, což je z pohledu výkonu nedostatečné a je určeno pouze pro testování. Doporučením je nastavit velikost na 50% z celkové velikosti RAM paměti[57]. V mém případě je nastaveno 8GB *-Xms8g -Xmx8g*. Otestovat funkčnost je možné pomocí tohoto příkazu. V případě, že je vrácena odpověď je ElasticSearch funkční.

```
curl -v http://localhost:9200
```

Výpis 9: Test funkčnosti ElasticSearch

10.2 Instalace Logstash

Jak již bylo zmíněno výše, Logstash se stará o sběr dat pro ElasticSearch. Pro svoji diplomovou práci zpracovávám vytvářené soubory s daty ohledně provozu, který jde přes reverse proxy. Je možné poskytnout jakýkoliv log, který následně dle konfigurace převede na formát JSON a odešle do ElasticSearche. Pro instalaci Logstash stačí pouze jeden příkaz:

```
sudo apt-get install logstash
```

Výpis 10: Příkaz pro instalaci Logstash

Konfigurační soubor */etc/logstash/conf.d/logstash-nginx-es.conf.save* obsahuje veškeré informace ohledně vstupu dat do Logstashe, jejich parsování a nastavení výstupu do ElasicSearche. Konkrétní soubor vypadá následovně.

```
input {
   beats {
      port => 5400
   }
}
filter {}
output {
   elasticsearch {
      hosts => ["localhost:9200"]
      index => "weblogs-%{+YYYY.MM.dd}"
      document_type => "nginx_logs"
   }
}
```

Výpis 11: Konfigurační soubor Logstash

V tomto konfiguračním souboru se nastavují vstupy, ze kterých Logstash sbírá data. V mém případě je nastaven vstup beats na port 5400. V sekci filter je možné definovat konkrétní způsob transformace dat pomocí groku[58]. Tento nástroj je ideální volbou na parsování jakéhokoliv formátu logu. Pro vytváření groku jsem využil doporučenou funkci dostupnou z *http://grokconstructor.appspot.com/*. Základní syntaxe je tento vzor *%SYNTAX:SEMANTIC*, kde *SYNTAX* je název vzoru který odpovídá textu v logu např. *NUMBER* a *SEMANTIC* určuje název hodnoty, jak bude uložena např. *response_code*. Grok vždy začíná zleva a proto je dobré při tvoření logu umístit data, které chceme logovat, na začátek. Ve filtru je také možné konkrétní parametry konvertovat na číselné hodnoty, aby dle nich bylo možné následně agregovat a vizualizovat v Kibaně. Další důležitý parametr v sekci filter je *date*. Na základě tohoto parametru je možné vytvořit časovou hodnotu, která může být použita jako časová osa. Typicky se jedná o datum a čas v záznamech serveru. Pokud tato hodnota není použita je defaultně použit čas odeslání požadavku do ElasticSearch a to může mít za následek zkreslení výsledných dat. Výsledný konfigurační soubor je umístěn v příloze. Logstash také poskytuje možnost zpracovávat JSON soubory pomocí TCP komunikace. Je tedy možné zpracovávat data z více serverů[59].

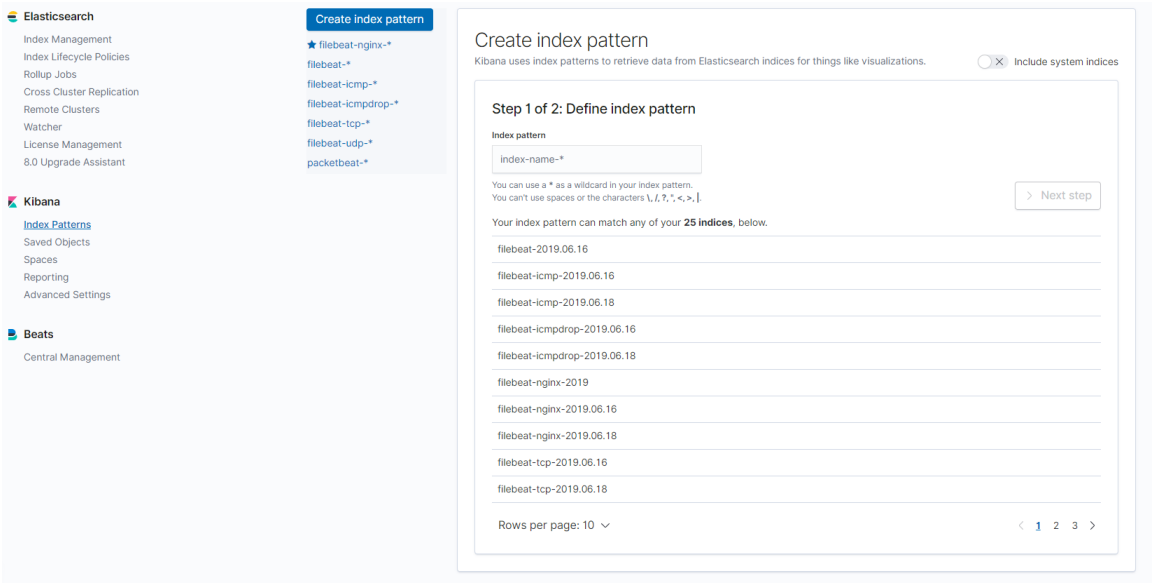
10.3 Instalace Kibana

Kibana je vizualizační nástroj, který dokáže z dat vytvořit časové, sloupcové, koláčové a další grafy. Také slouží k proházení všech vstupních dat ve formátu JSON. Kibana lze ovládat pomocí webového rozhraní. Instalace Kibany proběhne pomocí níže uvedeného příkazu.

```
sudo apt-get install kibana
```

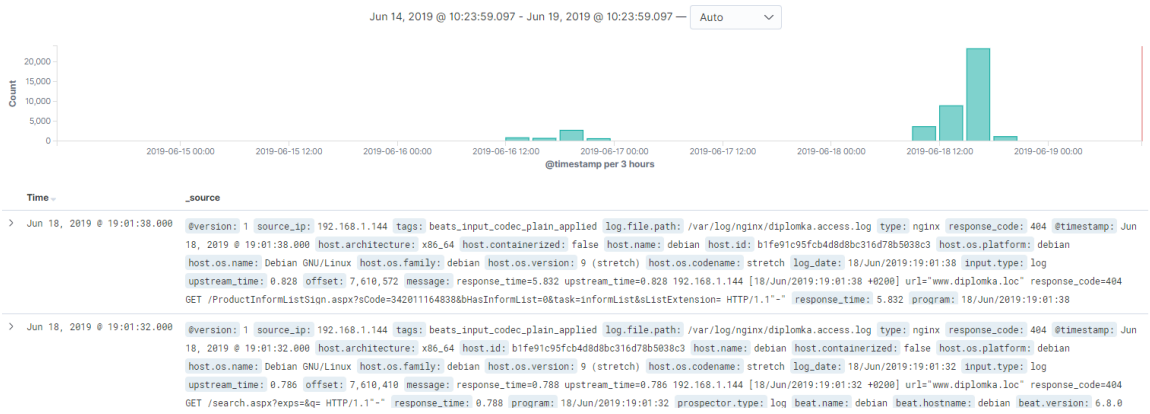
Výpis 12: Příkaz pro instalaci Kibana

Konfigurační soubor Kibany je umístěn v */etc/kibana/kibana.yml*. V tomto souboru je nutné nastavit server, na kterém bude Kibana umístěna konkrétně *server.host: "localhost"* a *server.port"5601*. Dále je také nutné nastavit URL adresu ElasticSearche *elasticsearch.url: "http://localhost:9200*
Po spuštění je Kibany je dostupná z dané URL adresy. Pro správnou funkčnost je nutné nastavit index patterns. Níže v obrázku je vidět sekce, kde se daná konfigurace provádí.



Obrázek 8: Nastavení indexů v Kibaně

Po vytvoření indexů je nyní možné prozkoumávat data přímo v Kibaně, v sekci Discovery. Níže v obrázku je možnost vidět vstupní data z logu Nginx, které následně v Kibaně vizualizuji.



Obrázek 9: Zobrazení dat v Kibaně

Kibana také poskytuje v základním nastavení pomocí služby *x-pack* monitoring daného serveru, nad kterým je kibana spuštěná. Toto zobrazení je možné nalézt v sekci *Monitoring*. Dokáže monitorovat např. vytíženost CPU, využití heapu a spoustu dalších informací.

10.4 Beats

K ELK stacku je možné připojit doplňkové služby tzv. Beatsy. Jedná se o software, který má na starost sbírání konkrétních souborů a dat. V diplomové práci používám FileBeat, který sbírá nadefinované soubory a odesílá je k parsování do Logstashe. Instalace FileBeatu probíhá pomocí těchto příkazů.

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key
    add -
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee
     -a /etc/apt/sources.list.d/elastic-7.x.list
sudo apt-get update
sudo apt-get install filebeat
```

Výpis 13: Instalace Filebeat

V konfiguračním souboru */etc/filebeat/filebeat.yml* je možné definovat cesty ke konkrétním souborům a spojení s Logstash. Pomocí níže uvedeného parametru *fields* je možné daný typ souboru označit, aby se pro každý log v Logstash používal jiný způsob parsování dat. Výsledný konfigurační soubor je umístěn v příloze této práce.

```
filebeat.inputs:
  -type: log
    enabled: true
    paths:
      - /var/log/nginx/diplomka.access.log
    fields: {log_type: nginx}
    exclude_files: ['\.gz$']
  -type: log
     enabled: true
     paths:
       - /var/log/icmp.log
     fields: {log_type: icmp}

output.logstash:
  hosts: ["localhost:5400"]
  worker: 2
```

Výpis 14: Konfigurace Filebeat

Dále jsou připraveny služby PacketBeat a MetricBeat pro budoucí použití.

PacketBeat[60] slouží k monitorování síťových protokolů, jako např. dns, http, tls a dalších. Packetbeat sbírá data a následně je odesílá do Logstash pro další zpracování nebo přímo do ElasticSearch. V Kibaně je následně možné nad těmito vytvářet vizualizace daných protokolů (portů).

MetricBeat[61] slouží k monitorování jednotlivých aplikací na daném serveru. Dokáže sbírat informace z aplikají jako např. Redis, MySQL, Apache, MongoDB a dalších.

11 Testovací prostředí

Pro diplomovou práci jsem od společnosti NetDirect s.r.o. dostal zapůjčený server společně se switchem *HP Procurve 1410-24G*. Tento server se standardně využívá v této firmě k testování nových funkcí a nastavení před uvedením do běžného provozu. Níže uvádím hardwarové vybavení serveru.

Procesor	2x Intel Xeon L5410 2.33GHz
Paměť	32 GB ECC
HDD	RAID10 15000rpm 240GB
Síťová karta	2x Intel(R) PRO/1000 EB
Síťová karta	2x Intel(R) PRO/1000 PT Server Adapter

Tabulka 2: Hardwarové vybavení serveru

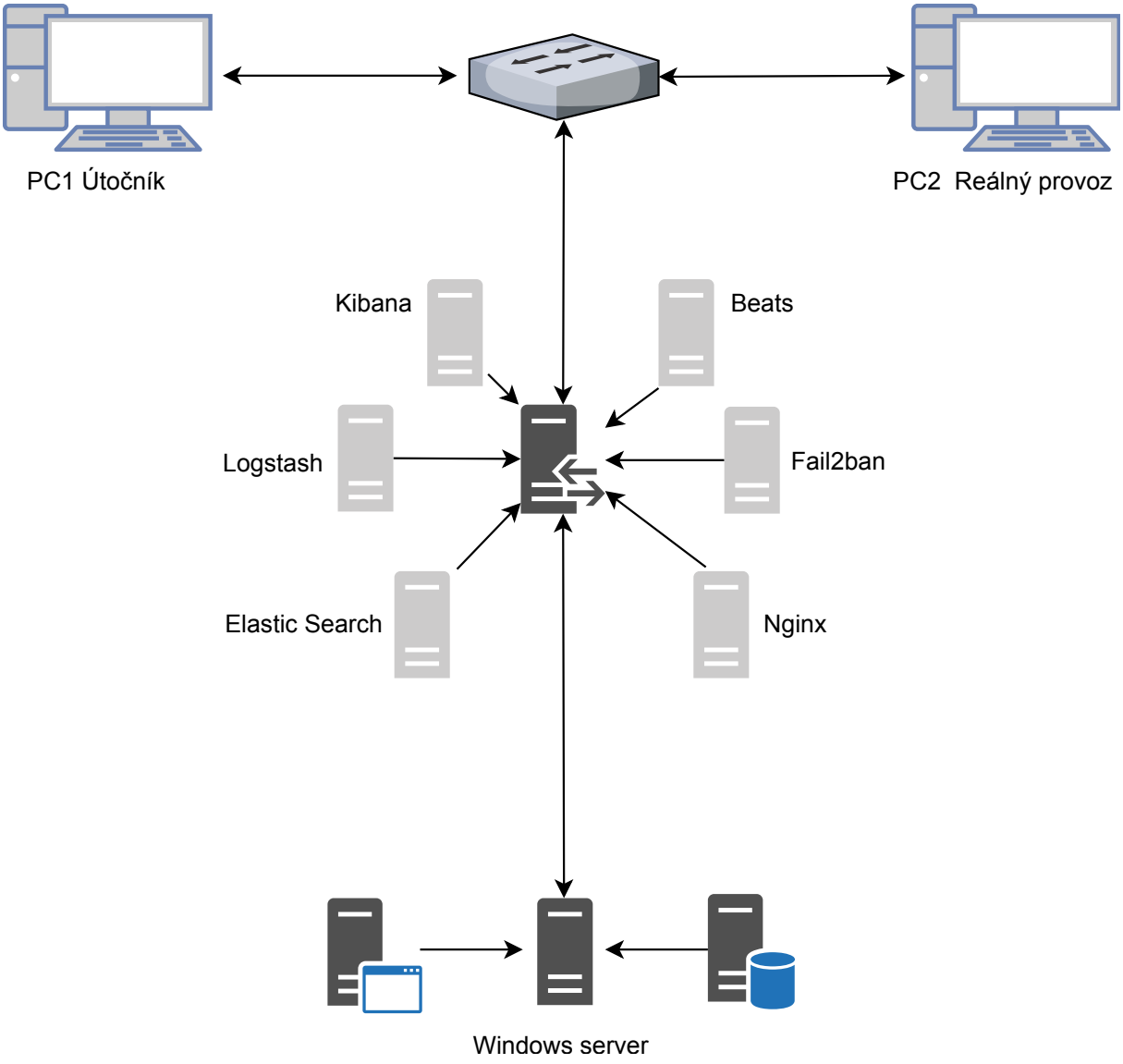
Na serveru je nainstalován operační systém Windows server 2008 R2. Na tomto serveru je funkční webová aplikace, poskytnutá firmou NetDirect s.r.o., na kterou bude simulován reálný provoz a prováděny útoky. Tato aplikace je postavena na ASP.NET. Databáze je založena na MSSQL. Vytvořil jsem pro testování virtuální stanici pomocí virtualizačního nástroje Hyper-V. Na této stanici je nainstalovaný operační systém Linux Debian9 v serverové verzi bez desktopového prostředí. Tato stanice slouží jako reverse proxy server a stará se o příchozí požadavky pro svého hosta. Na této stanici vytvářím analýzu a filtrování veškerého provozu, který reverse proxy server prochází. Tato stanice je také výstupem mé diplomové práce, kterou je možno použít pro jakýkoliv jiný server a tím zvýšit jeho bezpečnost proti útokům typu DoS/DDoS.

Útok bude prováděn pomocí PC, níže je uvedené hardwarové vybavení.

Procesor	Intel Core i5-4210H 2.90GHz
Paměť	8GB
OS	Kali Linux

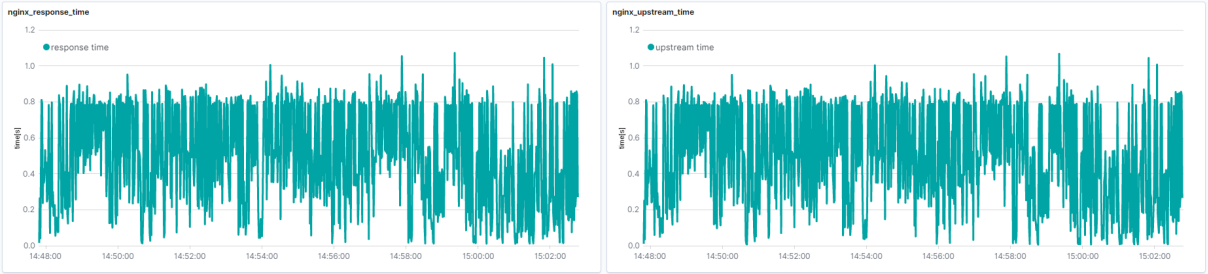
Tabulka 3: Hardwarové vybavení útočícího PC

Simulace reálného provozu probíhá na základě IIS logu dané webové aplikace poskytnuté firmou NetDirect s.r.o. pomocí programu Apache JMeter, který dokáže v čase vytvářet requesty z daného logu[62]. Daný log bylo nutné v prvé řadě transformovat do formátu, který dokáže Apache JMeter zpracovat. Pro zajištění šířky pásma jsem vytvořil pomocí *Intel Advanced Networking settings*[63] tým. Tato technologie umožňuje využít síťové adapetéry a spojit je dohromady, podobně jako vytvoření link aggregation u switche. Takto jsem dohromady spojil tři síťové adaptery, které jsou definované pouze pro reverse proxy server. Komunikace mezi reverse proxy a serverem následně probíhá přes poslední síťový adaptér. Níže v obrázku je zobrazeno celé zapojení testovacího prostředí pro lepší představu.



Obrázek 10: Testovací prostředí

Níže v grafech uvádím stav rychlosti odezvy webové aplikace v případě, že není pod žádným útokem. V levém grafu je zobrazena doba odezvy Nginx serveru, tedy celková doba od přijetí requestu, až po její odeslání zpět klientovi. V pravém grafu je zobrazena doba odezvy, mezi spojením Nginx a serveru na kterém daná webová aplikace běží.



Obrázek 11: Webová aplikace bez útoku

12 Testování konkrétních útoků

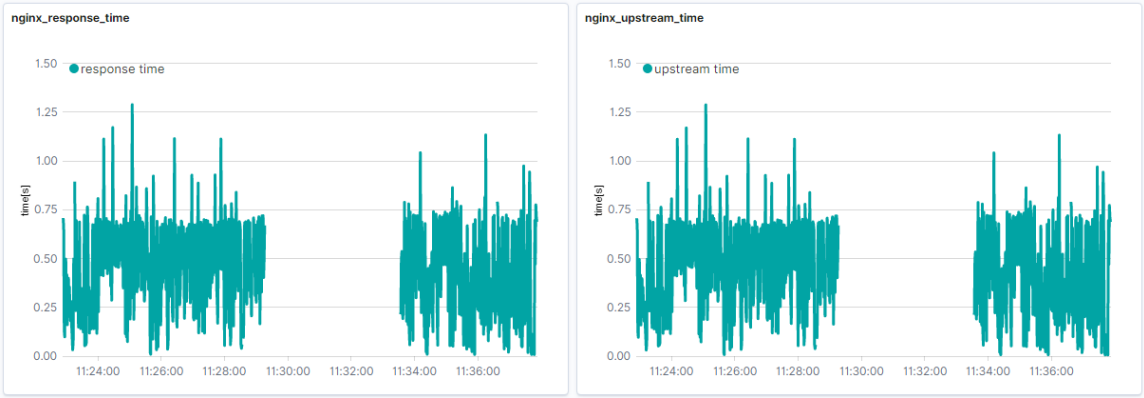
12.1 SYN Flood

Jak je již zmíněno výše, SYN Flood útok patří do kategorie Protocol Attacks. Snaží vytvořit co největší počet otevřených připojení na serveru, aby využil všechny jeho prostředky a znemožnil vytváření nových připojení pro běžné uživatele. Pro tento útok jsem využil funkci hping3, níže je uvedený konkrétní příkaz pro vytvoření SYN Flood útoku.

```
hping3 -S --flood 192.168.1.102 --rand-source
```

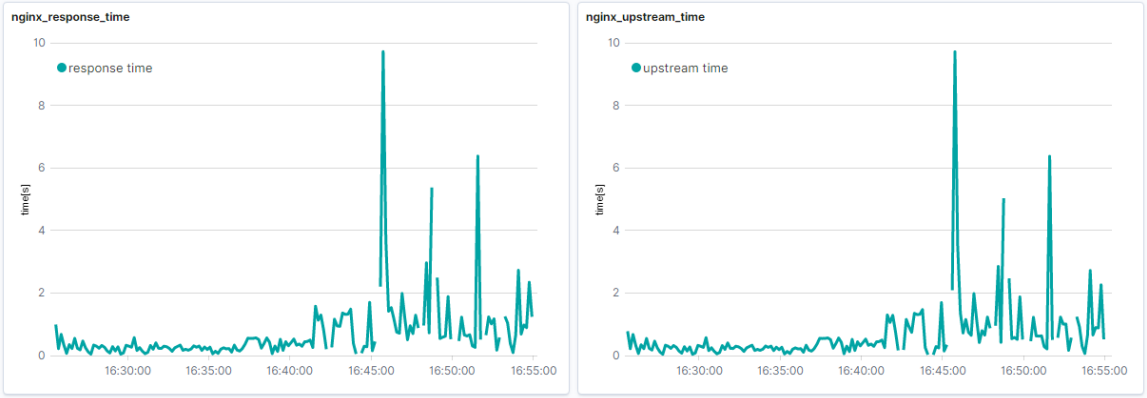
Výpis 15: SYN Flood příkaz pro útok

Zvolený parametr *-S* určuje použití SYN paketů. Tento příkaz byl celkem zapnut ve třech terminálech, kdy zatížení sítě na serveru dosáhlo cca 350Mbit/s. Po spuštění tohoto útoku došlo k okamžitému znemožnění vytváření spojení pro legitimní uživatele, DDoS útok byl tedy úspěšný. Níže v obrázku je vidět, že žádné připojení Nginx nezpracoval a webová aplikace byla nedostupná.



Obrázek 12: Aplikace pod SYN Flood útokem

Pro obranu proti tomuto útoku jsem využil TCP SYN cookies. Možnost tohoto nastavení je přímo v kernelu Debianu */etc/sysctl.conf* *net.ipv4.tcp_syncookies=1*, *net.ipv4.tcp_timestamps=1* a *net.ipv4.tcp_max_syn_backlog=16384*. Výše uvedené parametry zapnou SYN cookies, časovou značku, a nastaví maximální počet připojení. Maximální počet se odvíjí podle možnosti velikosti RAM paměti. Níže je uveden graf s vytvořenou obranou. Je vidět, že útok aplikaci zpomalil, ale nedokázal ji zastavit. Výše uvedená obrana je tedy proti tomuto útoku účinná.



Obrázek 13: Aplikace pod SYN Flood útokem s vytvořenou obranou

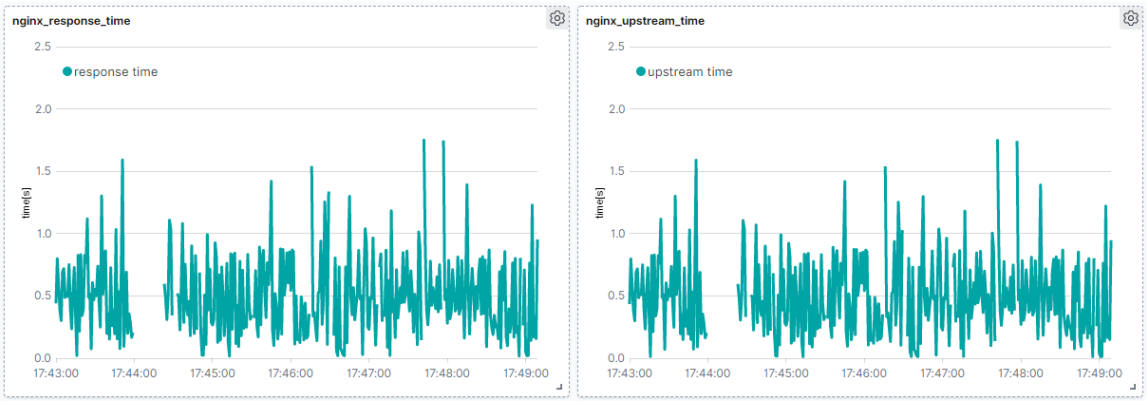
12.2 ICMP Flood

ICMP Flood útok patří do kategorie Volume Based Attacks. Cílem tohoto útoku je využít velkou šířku pásma cílového serveru a tím znemožnit přístup legitimním uživatelům. Pro vytvoření ICMP Flood útoku jsem použil funkci hping3, níže uvádím konkrétní příkaz ke spuštění daného útoku.

```
hping3 -1 --flood 192.168.1.102 -d 65000 --rand-source
```

Výpis 16: ICMP Flood příkaz pro útok

V daném příkazu je nutné zvolit parametr *-1* pro určení, že bude použit modul ICMP. Další parametr *-d* určuje velikost paketu v bajtech, zde konkrétně 65000 bajtů. Po spuštění byl vygenerován provoz v průměru cca 930 Mbit/s, tedy byla téměř využita celková kapacita útočníkovi linky. Níže v grafu jsou zaznamenány doby odezvy v čase během útoku. Je vidět, že ICMP Flood zásadně nezasáhl do běžného provozu a doba odezvy zůstala téměř nezměněna.



Obrázek 14: Server pod útokem ICMP Flood

Co by ovšem mohl být problém je odchozí provoz, který dosahoval až 850 Mbit/s. V případě, že by aplikace odesílala velké množství dat, např. nějaké exporty typicky datové feedy, mohlo by dojít k zahlcení dané linky, proto je potřeba daný odchozí provoz limitovat.

Pro limitaci ICMP provozu jsem použil pravidlo pro IPtables, kdy blokuji celý ICMP provoz. Běžný uživatel nikdy ICMP nevyužije, z toho důvodu je tedy zablokováno. Pro případné potřeby administrátorů sítě, je možné před dané pravidlo přidat povolení z daného rozsahu IP adres. Konkrétní příkazy vypadají následovně.

```
iptables -t mangle -A PREROUTING -p icmp -s xxx.xxx.xxx.xxx/24 -j ACCEPT
iptables -t mangle -A PREROUTING -p icmp -j DROP
```

Výpis 17: Příkazy k omezení ICMP provozu

Těmito opatřeními došlo ke snížení odchozího provozu až o 100% z původních 850Mbit/s k hodnotám okolo několika desítek Kbit/s. Těmito opatřeními je tedy útok úspěšně zmírněn a dále nemůže ovlivnit běžný provoz.

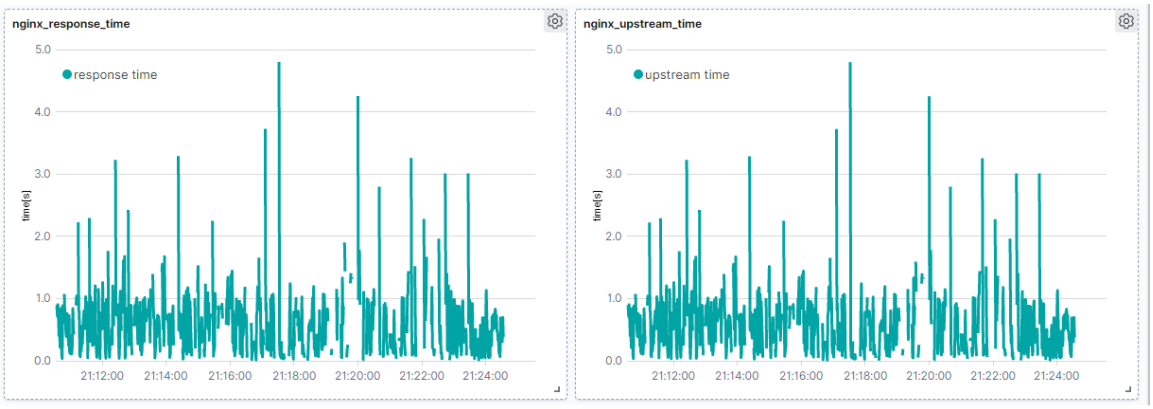
12.3 UDP Flood

Jak je již zmíněno výše, UDP Flood útok je velmi prostý, z důvodu, že se jedná o bezstavový protokol. Pro vygenerování UDP útoku jsem znovu použil základní funkci hping3. Níže uvádím konkrétní příkaz pro vygenerování daného útoku.

```
hping3 --udp --flood 192.168.1.102 --rand-source -p xx
```

Výpis 18: UDP Flood příkaz pro útok

Parametr *–udp* určuje, že bude použit UDP mód, jako další parametr je volitelný *–rand-source*, který definuje, že budou použity náhodné zdrojové IP adresy. Další parametr *-p* určuje, na který port bude prováděn útok. Pomocí daného příkazu bylo na server vygenerováno, za použití tří terminálů, cca 230Mbit/s provozu. Níže v grafu časů odezvy serveru je vidět, že během útoku vzrostla o více, než 300% oproti běžnému provozu. Útok tedy nezastavil celý provoz, ovšem ho citelně zpomalil.



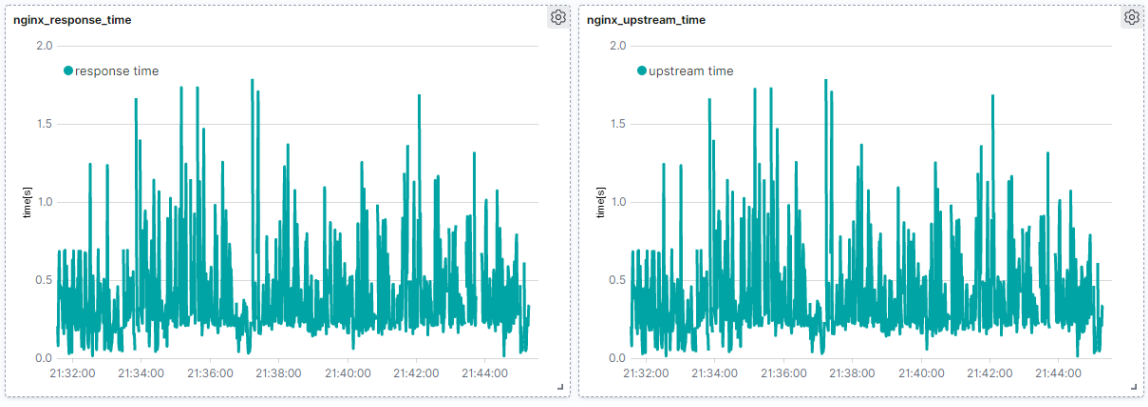
Obrázek 15: Server pod útokem UDP Flood

Pro běžný webový server není důvod mít UDP provoz zapnutý s výjimkou portu 53 pro DNS záznamy. Ochranu proti danému provozu tvořím pomocí IPtables, kde definuji celkovou blokaci UDP. Před daný záznam jsem také vložil omezení konkrétního portu UDP, tedy 53, na 100 požadavků za sekundu. Povolené i zablokované pakety následně loguji pro možnou vizualizaci dat v Kibaně. Níže jsou vypsány konkrétní pravidla.

```
iptables -A INPUT -p udp -dport 53 -m limit --limit 100/s --limit-burst 100 -j
    ACCEPTANDLOG
iptables -A INPUT -p udp -j DROPANDLOG
```

Výpis 19: Příkazy k omezení UDP provozu

Nastavením daných pravidel, byl pozitivní vliv na odezvu serveru. Klesla přibližně k původním hodnotám za běžného provozu. Níže v grafu je vidět citelný pokles doby odezvy všech požadavků. Díky těmto pravidlům je tedy UDP útok úspěšné zmírněn.



Obrázek 16: Server pod útokem UDP Flood s vytvořenou obranou

12.4 Slowloris

Slowloris se řadí do skupiny Low and Slow Attacks, které generují malé množství síťového provozu a útoční na aplikační vrstvu OSI modelu. Útok Slowloris se charakterizuje velkým množstvím vytvořených připojení, které ovšem nikdy nejsou dokončeny. Cílem je tedy vyčerpat maximální počet vytvořeným připojení. Pro tento útok jsem využil program napsaný v pythonu dostupný z *https://github.com/gkbrk/slowloris.git*. Jeho použití je velmi jednoduché, za to velmi účinné. Níže je uveden příkaz na spuštění toho programu.

```
python3 slowloris.py www.diplomka.loc
```

Výpis 20: Příkaz pro použití Slowloris

Pro tento útok jsem navíc použil parametr *-s*, který určuje počet soketů, které budou vytvořeny. Maximální počet, kterého jsem dosáhl bylo 1020. Daný útok měl za následek téměř okamžité zablokování provozu a tedy nedostupnosti celého systému. Níže je vidět výpis Nginx logu, který informuje o tom, že jsou využity všechny prostředky pro odbavování požadavků.

```
2019/06/20 02:04:46 [alert] 2294#2294: 768 worker_connections are not enough
2019/06/20 02:04:46 [alert] 2294#2294: 768 worker_connections are not enough
2019/06/20 02:04:46 [alert] 2294#2294: 768 worker_connections are not enough
2019/06/20 02:04:46 [alert] 2294#2294: 768 worker_connections are not enough
2019/06/20 02:04:53 [alert] 2294#2294: 768 worker_connections are not enough
2019/06/20 02:04:55 [alert] 2294#2294: 768 worker_connections are not enough
```

Obrázek 17: Server pod útokem Slowloris

Pro obranu jsem se rozhodl využit IPtables, kdy blokuji maximální počet vytvořených připojení z jedné IP adresy. Pro otestování jsem povolil veškerý simulovaný reálný provoz z IP adresy. Následně je přidáno níže zmíněné pravidlo. Konkrétní tvar pravidla je následující.

```
iptables -A INPUT -tcp -m connlimit --connlimit-above 100 -j DROPANDLOG
```

Výpis 21: Omezení TCP připojení

Tento příkaz povoluje maximální počet TCP připojení pro jednu IP adresu na 100. V případě překročení tohoto limitu je daná informace zaznamenána a paket je zahozen.

Výsledkem tohoto testu je úspěšné zablokování nežádoucího provozu, který je před tím zalogován. V případě je možné použít funkce File2ban, která dokáže daný log pomocí regexu přečíst a blokovat příchozí IP adresy. Níže je vidět, stav generování útoku po nasazení daného pravidla, kde z původního počtu 1020 soketů okamžitě spadlo na zmíněných 100. Jako další možnost obrany je vytvoření bash skriptu, který bude automaticky přidávat záznamy do IPtables na základě výpisu z netstatu. Tento script může vypadat následovně.

```
#!/usr/bin/env sh
netstat -an |
```

```
awk -vmax=100 '/tcp/{split($5,a,":"); if(a[1] > 0 && a[1]!="0.0.0.0"){c[a
    [1]]++}}
END{for(ip in c){if(c[ip]>max){print ip}}}' |
while read ip; do iptables -t mangle -I PREROUTING 1 -s "$ip" -j DROP; done
```

Výpis 22: Script pro blokaci IP adres

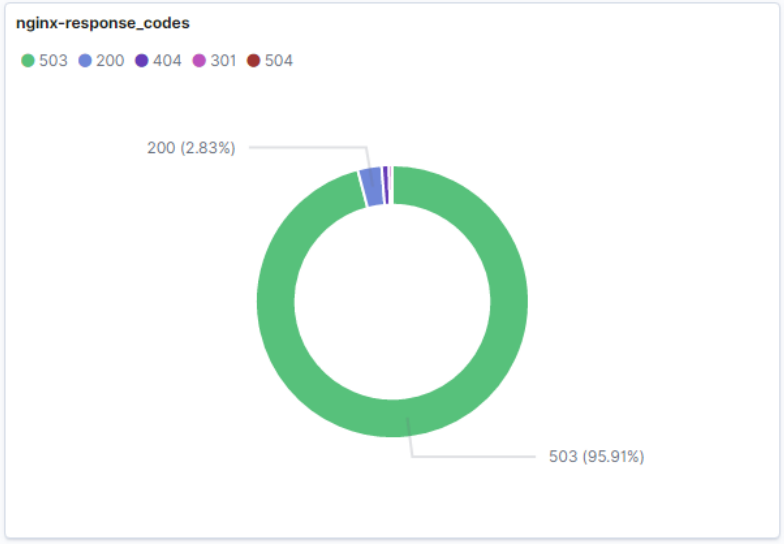
S umístěním do cronu může být skript spouštěn např. každou minutu a tím daný útok aktivně blokovat. V nastavení kernelu systému je možné nastavit volbu *net.core.somaxconn*, pro zvýšení maximální počtu současně vytvořených spojení. Tato volba je velmi důležitá v případě často využívaného serveru.

```
root@heinzz:/home/heinzz/wreckuests/slowloris# python3 slowloris.py www.diplomka
.loc -s 5000
[20-06-2019 02:04:09] Attacking www.diplomka.loc with 5000 sockets.
[20-06-2019 02:04:09] Creating sockets...
[20-06-2019 02:04:10] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:04:26] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:04:41] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:04:56] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:05:11] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:05:26] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:05:41] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:05:56] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:06:11] Sending keep-alive headers... Socket count: 1020
[20-06-2019 02:08:15] Sending keep-alive headers... Socket count: 100
[20-06-2019 02:08:34] Sending keep-alive headers... Socket count: 100
[20-06-2019 02:08:53] Sending keep-alive headers... Socket count: 100
[20-06-2019 02:09:12] Sending keep-alive headers... Socket count: 100
[20-06-2019 02:08:53] Sending keep-alive headers... Socket count: 100
[20-06-2019 02:09:12] Sending keep-alive headers... Socket count: 100
```

Obrázek 18: Server pod útokem Slowloris s vytvořenou obranou

12.5 HTTP Flood

Tento útok patří do kategorie Application Layer Attacks a útočí přímo na webovou aplikaci. Podobně jako u Slowloris útoku, se snaží HTTP Flood vytvořit velké množství požadavků na cílový server s rozdílem, že se jedná o útok hrubou silou s následkem využití všech možných prostředků a tím znemožnění odbavování všech ostatních. Pro tento útok jsem využil LOIC, který popisuji v kapitole Nástroje pro útok. S tímto programem se mi podařilo vygenerovat 150000 požadavků během tří minut. Následkem tohoto útoku bylo okamžité zastavení funkčnosti webové aplikace. Níže uvádím graf odpovědí serveru, kde je vidět, že de facto všechny requesty končí stavem 503.



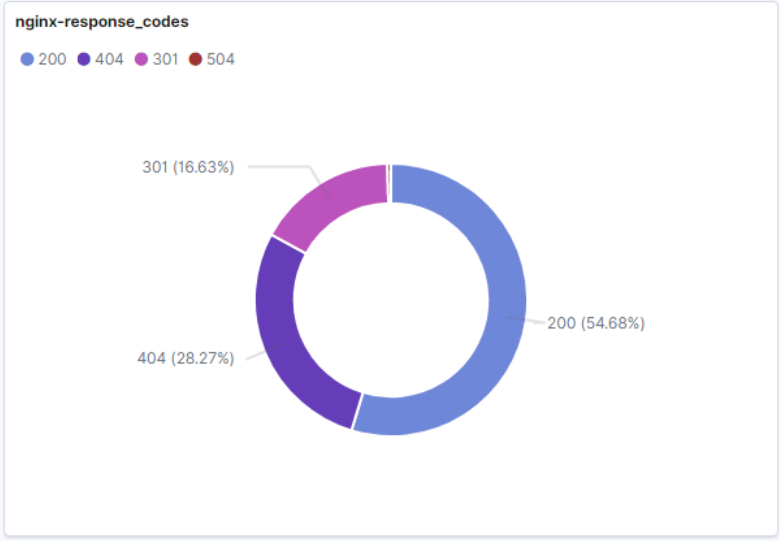
Obrázek 19: Server HTTP Flood útokem

Pro obranu proti tomuto útoku jsem v první řadě použil nastavení IPtables stejné jako v případě Slowloris útoku, tedy blokaci maximálního počtu vytvořených připojení z jedné IP adresy na 100.

Jako druhou možnost jsem vyzkoušel daný bash skript, který automaticky doplňuje blokaci IP adres na základě výpisu z netstat a umísťuje je do IPtables. Nevýhodou tohoto skriptu je ovšem to, že daná IP adresa je blokovaná do restartu serveru nebo do ručního odstranění pravidla kompetentním administrátorem. Výsledek je ovšem stejný, jako v případě první možnosti.

Třetí možností, kterou jsem použil bylo využití programu *File2ban*, který na základě zadaného log souboru a další pravidel, které popisuji v další kapitole, blokuje IP adresy pomocí IPtables. Tato možnost mi z výše uvedených připadá, jako nejlepší způsob, jak automaticky generovat pravidla pro IPtables.

Výsledkem všech výše zmíněných možností je úspěšná blokace daného provozu. Výsledek je možné vidět níže v grafu, kde je zobrazeno, že provoz se vrátil k normálu, pouze minimální počet požadavků skončil chybou 504, tedy time-outem.



Obrázek 20: Server HTTP Flood útokem s vytvořenou obranou

12.6 Obrana pomocí File2ban

File2Ban je služba, která na základě přiřazených souborů dokáže automaticky blokovat a vytvářet pravidla pro IPtables.

Tuto funkci jsem použil z důvodu velké škálovatelnosti, kdy je možné vytvořit pravidlo pro de facto jakýkoliv log, který obsahuje IP adresu. Tímto programem je možné chránit např. SSH server, FTP server, DNS server a mnoho dalších, v mém případě právě Nginx server.

Další velkou výhodou tohoto programu je možné nastavení, za jakých pravidel bude daná IP adresa blokována a na jak dlouhou dobu. Toto je hlavní důvod, proč jsem tento program použil, protože je bezúdržbový. Instalace programu probíhá pomocí tohoto příkazu:

```
apt-get install fail2ban
```

Výpis 23: Instalace Fail2ban

Nyní je nutné editovat konfigurační soubor */etc/fail2ban/jail.conf*. V tomto souboru je možné vybrat defaultní nastavení, ve kterém lze nastavit ignoraci zvolených IP adres *ignoreip = xxx.xxx.xxx.xxx/yy*. Je možné vložit více rozsahů, stačí dané adresy oddělit mezerou. V konfiguraci je možné nalézt již předdefinované nastavení daných služeb. Zde se zaměřím na konkrétní, kterou používám a to *[nginx-limit-req]*. Níže uvádím nastavení, které jsem použil.

```
port    = http
logpath = /var/log/nginx/diplomkafile2ban.log
enabled = true
findtime = 600
bantime = 3600
maxretry = 10
```

```
action   = iptables-multiport[name=ReqLimit, port="http", protocol=tcp]
filter   = nginx-limit-req
```

Výpis 24: Konfigurace File2ban

Toto zmíněné nastavení určuje cestu k log souboru, ze kterého budou brány informace. Další důležitý parametr je *findtime*, který určuje dobu, po kterou běží jeden měřící cyklus. Dalším parametrem je *bantime*, který určuje dobu, po kterou bude daná IP adresa zablokována. Parametr *maxretry* počet záznamů v logu. Pokud tento počet je během času *findtime* překročen, je daná IP adresa blokována pomocí definovaného parametru *action*. *Filter* určuje název konfiguračního souboru, který bude použit. Tento konfigurační soubor je možné najít ve složce */etc/fail2ban/filter.d*. Konfigurační soubor pro Nginx je vytvořen již v základu.

Pro funkčnost File2ban ve spojení s Nginx je nutné použít *nginx limit req module*[64]. Bylo nutné změnit konfiguraci samotného Nginx, aby tento modul začal využívat. Konfigurace je možná v */etc/nginx/nginx.conf*, zde jsem přidal parametr *limit_req_zone $binary_remote_addr zone=one:10m rate=100r/s;*. Tento parametr určuje název zóny, velikost v MB a počet požadavků za sekundu. Velikost v MB určuje maximální počet uložených IP adres. 1MB dokáže obsloužit 16000 IP adres[64]. Nyní je možné přidat do */etc/nginx/sites-enabled/reverse-proxy.conf* parametr, který určuje, že pro daný server bude pravidlo využíváno *limit_req zone=one*. Tento parametr je možné použít pro konkrétní server a také pro nastavené lokace, např. danou funkci na webu bude možné používat pouze v omezeném režimu. Navíc jsem definoval error log, do kterého se záznamy ukládají. Stačí přidat *error_log /var/log/nginx/diplomkafile2ban.log;* do nastavení konkrétního serveru.

File2ban provedené akce, ohledně nalezení, zablokování a odblokování nebezpečné IP adresy, ukládá do svého log souboru. Tento soubor je možné pomocí Filebeatu sbírat pro Logstash a následně data zaslat do ElasticSearche k vizualizaci v Kibaně. Touto možností je mít vždy rychlý přehled o počtu zachycených incidentů a popř. provést adekvátní kroky k ošetření tohoto problému.

13 Výsledky testování

Testováno bylo celkem 5 útoků, na které byla úspěšně vytvořena obrana a útoky byly zmírněny.

1. TCP SYN Flood

   Pro obranu proti tomuto útoku jsem využil nastavení kernelu linuxu. Zapnutím syn_cookies a nastavením většímu počtu TCP spojení došlo k zásadnímu zmírnění tohoto útoku, z původního stavu, kdy byla aplikace nedostupná do stavu, kdy byla "pouze"zpomalená. Jako další možností obrany proti tomuto útoku je možné využít RST cookies, kdy je uživateli jako odpověď na SYN request odeslán SYN-ACK se špatným sekvenčním číslem. Legitimní zákazník na tento požadavek musí odpovědět pomocí RST.

2. UDP Flood

   Proti tomuto útoku jsem využil limitaci příchozích požadavků pro port 53 a zablokování všech ostatních pomocí IPtables. Bez ochrany tento útok způsobil citelné zpomalení webové aplikace. Po nasazení dané obrany se vše vrátilo téměř do normálu.

3. ICMP Flood

   Tento útok z prvního pohledu byl neúspěšný, aplikace zpomalena nebyla. Jediný příznak útoku bylo velké zatížení sítě, který by mohl ohrozit provoz v případě špičky návštěvnosti během sezóny daného webové aplikace např. Vánoce. Pro limitování tohoto útoku jsem použil nastavení IPtables, kdy všechen ICMP provoz blokuji. Tímto opatřením došlo k enormnímu snížení odchozího pásma. Takto je zajištěno, že v případě velkého množství odesílaných dat, bude server připraven.

4. Slowloris

   Útok tohoto typu velmi rychle zablokoval celkový provoz a nešel vytvořit žádný požadavek směrem k dané webové aplikaci. Pro zamezení tohoto útoku jsem využil nastavení IPtables, kdy limituji příchozí TCP provoz z jedné IP adresy na 100. Tímto omezením nebyl překročen maximální počet vytvořených spojení a tedy byl prostor pro odbavování běžného provozu.

5. HTTP Flood

   Proti tomuto útoku jsem testoval tři způsoby obrany, které dosáhli stejného výsledku a to zablokováním nežádoucího provozu. Bez obrany tento útok způsobil velmi rychlý pád celé aplikace, všechny requesty končili stavem 503. Tato obrana je ovšem pouze základní, funguje pouze na základě počtu požadavků z dané IP adresy. Pro vylepšení této metody navrhuji využít způsob měření reputace IP adresy, např. na základě běžného počtu požadavků, dokončení cíle webu např. zakoupení zboží na e-shopu a dalších faktorů.

Pro všechny tyto útoky je možné využít připravený bash script, který měří provoz a na základě toho blokuje dané IP adresy, popř. jejich rozsahy.

Doporučuji využít funkce File2ban, která je popsána v předchozí kapitole, která na základě logů dokáže dané IP adresy blokovat po určitou dobu a následně opět povolit. Dále doporučuji v případě použití jakéhokoliv z výše uvedených opatření vždy pečlivě zanalyzovat provoz směřující k webové aplikaci a nastavení daných pravidel dle konkrétních potřeb každé aplikace, aby nedošlo k nežádoucímu zablokování legitimních uživatelů. Na základě výsledků si dovolím říct, že vybrané a nasazené způsoby obrany jsou schopny detekovat nežádoucí provoz a úspěšně zmírňovat dopady těchto útoků.

13.1 Rozšíření obrany

Na základě výsledků útoků bych navrhoval rozšířit obranu proti útoku HTTP Flood a jeho alternativám, jako Single Request HTTP Flood, na základně reputace IP adres. Útok typu Single Request HTTP Flood, který popisuji v teoretické části této práce může být pro danou webovou aplikaci hrozbou. Útočník provede zkoumání systému obrany a zjistí kolik požadavků server zpracuje, než bude zablokován. Použitím botnetu dojde k vygenerování velkému počtu požadavků, které server nemusí být schopem obsluhovat.

Jako příklad použiji e-shopovou aplikaci, kterou jsem měl k těmto účelům zapůjčenou. Cílem e-shopu je prodat uživateli zboží. Standartně při nákupu zákazník dělá proces, který je podobný ostatním zákazníkům. Zákazník prochází kategorie e-shopu, používá vyhledávání, zobrazuje si detaily produkty a čte si jejich popis/parametry. Následně vloží produkt do košíku a pokračuje nákupním procesem. Na základě těchto skutečností je možné vytvořit scénaře běžného provozu e-shopu i za pomocí třetích stran, např. Google analytics. Chování uživatelů je možné klasifikovat a použít pro obranu proti těmto typům útoků. Do budoucna také pokrýt větší množství známých útoků, jako je RST Flood, R.U.Dead Yet? a dalších.

14 Závěr

V teoretické části této práce v prvé řadě popisuji a vysvětluji DoS/DDoS útoky a způsoby útoků v dnešním internetu pomocí botnetu. Následně popisuji stav dnešní obrany, četnosti těchto útoků a konkrétní pohledy firem, které se zabývají útoky typu DDoS. Na základě analýzy firem bylo vybráno 5 útoků, kterým se dále více věnuji. V další kapitole popisuji konkrétní typy útoků, jak se používají a na co cílí. V kapitole Obrana proti DoS/DDoS popisuji teoretické způsoby, jak přistupovat k obraně proti těmto útokům a konkrétní způsoby detekce a filtrace jednotlivých útoků.

Cílem praktické části bylo vytvořit nástroj pro analýzu a filtraci nežádoucího provozu typu DoS/DDiS. Pro účely této práce bylo využito reverse proxy serveru, který se o analýzu a filtraci stará a propouští pouze legitimní provoz.

V kapitole Nástroje pro útok popisuji programy hping3, LOIC a Slowloris.py, které jsem ke generování útoků použil. Dále je popsán konkrétní způsob realizace pomocí aplikací Nginx reverse proxy, ELK stacku a Filebeatu, jejich nastavení a způsob využití, jako jeden celek pro analýzu síťového provozu. Všechny tyto nástroje byly nainstalovány na operační systém Linux Debian9, který byl realizován pomocí virtualizačního nástroje Hyper-V. Na hostujícím fyzickém serveru byla spuštěna reálná e-shopová aplikace, která je denně využívána několika tisíci uživateli pro nákup zboží. Na tuto aplikaci byly dané útoky typu DoS/DDoS směřovány. Testovací server a webovou aplikaci jsem měl zapůjčenou od firmy NetDirect s.r.o.

V další části popisuji způsob vlivu jednotlivých útoku na danou webovou aplikaci a následnou obranu proti každému z těchto útoku. Všechny tyto vlivy útoků vizualizuji pomocí ELK stacku, kde jsou v grafech názorně zobrazené dopady na danou webovou aplikaci. K jednotlivým útokům popisuji konkrétní způsob nasazené obrany a vizualizuji účinnost těchto opatření. Na konci testování shrnuji zjištěné výsledky a přidávám doporučení pro uživatele, kteří budou chtít podobnou obranu aplikovat na svoje řešení, společně s možností rozšíření obrany proti DoS/DDoS útokům.

Během tvorby této diplomové práce jsem se více seznámil s danými typy útoků a způsobu jejich zamezení. Sám jsem se znovu přesvědčil, že DoS/DDoS útoky jsou v dnešní době stále reálnou hrozbou pro internetové služby a je třeba se jimi zabývat.

Bc. Jan Novotný

Literatura

[1] 15 Most Dangerous DDoS Attacks That Ever Happened [online]. June 23, 2016 [cit. 2019-06-12]. Dostupné z: https://www.globaldots.com/15-most-dangerous-ddos-attacks-that-ever-happened/

[2] New York's Panix Service Is Crippled by Hacker Attack [online]. September 14, 1996 [cit. 2019-06-12]. Dostupné z: https://archive.nytimes.com/www.nytimes.com/library/cyber/week/0914panix.html

[3] Denial of Service [online]. February 3, 2015 [cit. 2019-06-12]. Dostupné z: https://www.owasp.org/index.php/Denial_of_Service

[4] DDoS attacks in Q4 2017 [online]. February 6, 2018 [cit. 2019-06-12]. Dostupné z: https://securelist.com/ddos-attacks-in-q4-2017/83729/

[5] Za výpadky volebních webů statistického úřadu stojí hackerský útok [online]. 22. října 2017 [cit. 2019-06-12]. Dostupné z: https://www.irozhlas.cz/volby/parlamentni-volby-2017-csu-scitani-hlasu-hackersky-utok_1710221812_sam

[6] What is a botnet? [online]. [cit. 2019-06-12]. Dostupné z: https://us.norton.com/internetsecurity-malware-what-is-a-botnet.html

[7] DDoS in the IoT: Mirai and Other Botnets [online]. 07 July 2017 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/7971869

[8] DDoS Attacks in Q3 2018 [online]. [cit. 2019-06-23]. Dostupné z: https://securelist.com/ddos-report-in-q3-2018/88617/

[9] What is a Botnet? [online]. [cit. 2019-06-12]. Dostupné z: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-botnet/

[10] DDOS ATTACKS: 3 COMMON MOTIVATIONS [online]. [cit. 2019-06-23]. Dostupné z: https://www.trustedknight.com/ddos-attacks-3-common-motivations/

[11] The cost of launching a DDoS attack [online]. March 23, 2017 [cit. 2019-06-21]. Dostupné z: https://securelist.com/the-cost-of-launching-a-ddos-attack/77784/

[12] Booters, Stressers and DDoSers [online]. [cit. 2019-06-21]. Dostupné z: https://www.imperva.com/learn/application-security/booters-stressers-ddosers/

[13] Criminal Benefits: Profit Margin of a DDoS Attack Can Reach 95% [online]. [cit. 2019-06-23]. Dostupné z: https://www.kaspersky.com/about/press-releases/2017_criminal-benefits

[14] Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment [online]. 01 June 2015 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/7113475

[15] DDoS Protection Service | Anti DDoS Mitigation | Cloudflare [online]. [cit. 2019-06-21]. Dostupné z: https://www.cloudflare.com/ddos/

[16] DDoS Attacks in Q4 2018. SecureList [online]. [cit. 2019-05-29]. Dostupné z: https://securelist.com/ddos-attacks-in-q4-2018/89565/

[17] A Look Back At The DDoS Trends of 2018 [online]. January 28, 2019 [cit. 2019-06-11]. Dostupné z: https://blogs.akamai.com/2019/01/a-look-back-at-the-ddos-trends-of-2018.html

[18] NSFOCUS Identifies DDoS Attack Trends in New 2018 Insights Report - NSFOCUS, Inc., a global network and cyber security leader, protects enterprises and carriers from advanced cyber attacks. [online]. April 10, 2019 [cit. 2019-06-12]. Dostupné z: https://nsfocusglobal.com/nsfocus-identifies-ddos-attack-trends-new-2018-insights-report/

[19] Cybersecurity threatscape 2018: trends and forecasts [online]. [cit. 2019-06-21]. Dostupné z: https://www.ptsecurity.com/ww-en/analytics/cybersecurity-threatscape-2018/

[20] DDoS Threat Report 2018 Q4 [online]. [cit. 2019-06-21]. Dostupné z: https://www.nexusguard.com/threat-report-q4-2018

[21] DDoS Threat Report 2018 Q2 [online]. [cit. 2019-06-21]. Dostupné z: https://www.nexusguard.com/threat-report-q2-2018

[22] Choosing a DDoS mitigation solution ... the cloud based approach [online]. June 10, 2013 [cit. 2019-06-23]. Dostupné z: https://www.cyberdefensemagazine.com/choosing-a-ddos-mitigation-solution-the-cloud-based-approach/

[23] Low and Slow Attack [online]. [cit. 2019-06-23]. Dostupné z: https://www.cloudflare.com/learning/ddos/ddos-low-and-slow-attack/

[24] UDP - Flood [online]. [cit. 2019-06-20]. Dostupné z: https://www.f5.com/services/resources/glossary/udp-flood

[25] Contiki-based mitigation of UDP flooding attacks in the Internet of things [online]. 21 December 2017 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/8229997

[26] Threat - Report [online]. [cit. 2019-06-23]. Dostupné z: https://www.nexusguard.com/hubfs/Threat%20Report%20Q2%202018/ Nexusguard_DDoS_Threat_Report_Q2_2018_EN.pdf

[27] How to Perform DDoS Test as a Pentester [online]. December 3, 2016 [cit. 2019-06-19]. Dostupné z: https://pentest.blog/how-to-perform-ddos-test-as-a-pentester/

[28] Design of TCP SYN Flood DDoS attack detection using artificial immune systems [online]. 13 February 2017 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/7849626

[29] Design and Implementation of an OpenFlow-Based TCP SYN Flood Mitigation [online]. 30 April 2018 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/8350436

[30] DDoS attack algorithm using ICMP flood [online]. 31 October 2016 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/7725026

[31] ICMP - ECHO / ECHO REPLY (PING) MESSAGE [online]. [cit. 2019-06-12]. Dostupné z: http://www.firewall.cx/networking-topics/protocols/icmp-protocol/152-icmp-echo-ping.html

[32] What is a Slowloris DDoS attack? [online]. [cit. 2019-06-19]. Dostupné z: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/

[33] Útok Slowloris aneb plíživé nebezpečí pro web servery [online]. 17. 5. 2011 [cit. 2019-06-20]. Dostupné z: https://www.root.cz/clanky/utok-slowloris-aneb-plizive-nebezpeci-pro-web-servery

[34] What is a R.U.D.Y. attack? [online]. [cit. 2019-06-21]. Dostupné z: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/r-u-dead-yet-rudy/

[35] 35 Types of DDoS Attacks Explained [online]. [cit. 2019-06-21]. Dostupné z: https://javapipe.com/blog/ddos-types/

[36] Impact Analysis of HTTP and SYN Flood DDoS Attacks on Apache 2 and IIS 10.0 Web Servers [online]. 29 November 2018 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/8548783

[37] What is an HTTP flood attack [online]. [cit. 2019-06-12]. Dostupné z: https://www.imperva.com/learn/application-security/http-flood/

[38] Application Denial of Service [online]. 22 April 2010 [cit. 2019-06-12]. Dostupné z: https://www.owasp.org/index.php/Application_Denial_of_Service

[39] An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment [online]. 06 April 2017 [cit. 2019-06-12]. Dostupné z: https://ieeexplore.ieee.org/document/7893798

[40] What is an HTTP flood attack [online]. [cit. 2019-06-22]. Dostupné z: https://www.imperva.com/learn/application-security/http-flood/

[41] How to Set a Ping Packet Size [online]. [cit. 2019-06-12]. Dostupné z: https://www.techwalla.com/articles/how-to-set-a-ping-packet-size

[42] Transmission Control Protocol [online]. 11 June 2019 [cit. 2019-06-12]. Dostupné z: https://en.wikipedia.org/wiki/Transmission_Control_Protocol

[43] TCP SYN Cookies – DDoS defence [online]. 12 SEPTEMBER 2008 [cit. 2019-06-12]. Dostupné z: https://etherealmind.com/tcp-syn-cookies-ddos-defence/

[44] What Is A Reverse Proxy? | Proxy Servers Explained [online]. [cit. 2019-06-23]. Dostupné z: https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/

[45] UDP Flood Attack [online]. [cit. 2019-06-23]. Dostupné z: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/

[46] IDS vs. IPS: What is the Difference? [online]. [cit. 2019-06-23]. Dostupné z: https://www.varonis.com/blog/ids-vs-ips/

[47] Iptables(8) - Linux man page [online]. [cit. 2019-06-23]. Dostupné z: https://linux.die.net/man/8/iptables

[48] Iptables-extensions [online]. [cit. 2019-06-23]. Dostupné z: http://ipset.netfilter.org/iptables-extensions.man.html

[49] Hping3 Package Description [online]. [cit. 2019-06-22]. Dostupné z: https://tools.kali.org/information-gathering/hping3

[50] Low Orbit Ion Cannon [online]. [cit. 2019-06-19]. Dostupné z: https://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon

[51] LOIC (Low Orbit Ion Cannon) – DOS attacking tool [online]. [cit. 2019-06-23]. Dostupné z: https://resources.infosecinstitute.com/loic-dos-attacking-tool/

[52] What is NGINX? [online]. [cit. 2019-06-21]. Dostupné z: https://www.nginx.com/resources/glossary/nginx/

[53] Usage statistics of Nginx [online]. [cit. 2019-06-21]. Dostupné z: https://w3techs.com/technologies/details/ws-nginx/all/all

[54] ELK Stack: Elasticsearch, Logstash, Kibana | Elastic [online]. [cit. 2019-06-19]. Dostupné z: https://www.elastic.co/elk-stack

[55] These 15 Tech Companies Chose the ELK Stack [online]. [cit. 2019-06-21]. Dostupné z: https://logz.io/blog/15-tech-companies-chose-elk-stack/

[56] Elasticsearch: RESTful, Distributed Search & Analytics | Elastic [online]. [cit. 2019-06-19]. Dostupné z: https://www.elastic.co/products/elasticsearch

[57] Setting the heap size [online]. [cit. 2019-06-19]. Dostupné z: https://www.elastic.co/guide/en/elasticsearch/reference/current/heap-size.html

[58] Grok filter plugin [online]. [cit. 2019-06-20]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html

[59] Tcp input plugin [online]. [cit. 2019-06-22]. Dostupné z: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-tcp.html

[60] Packetbeat Reference [online]. [cit. 2019-06-23]. Dostupné z: https://www.elastic.co/guide/en/beats/packetbeat/current/index.html

[61] Metricbeat Reference [online]. [cit. 2019-06-23]. Dostupné z: https://www.elastic.co/guide/en/beats/metricbeat/current/index.html

[62] Access Log Sampler [online]. [cit. 2019-06-19]. Dostupné z: https://jmeter.apache.org/usermanual/component_reference.html#Access_Log_Sampler

[63] Advanced Driver Settings for Intel® Ethernet 10 Gigabit Server Adapters [online]. 03/25/2019 [cit. 2019-06-19]. Dostupné z: https://www.intel.com/content/www/us/en/support/articles/000005783/network-and-i-o/ethernet-products.html

[64] Rate Limiting with NGINX and NGINX Plus [online]. June 12, 2017 [cit. 2019-06-20]. Dostupné z: https://www.nginx.com/blog/rate-limiting-nginx/

Přílohy

Součástí této práce je CD, na kterém naleznete:

1. Konfigurační soubor ElasticSearch

2. Konfigurační soubor Logstash

3. Konfigurační soubor Kibana

4. Konfigurační soubor Filebeat

4. Konfigurační soubor Nginx

5. Konfigurační soubor Nginx-reverse proxy

6. Konfigurační soubor Fail2ban

7. Soubor s IPtables

8. Konfigurační soubor kernelu